TRITON LEVEL SIX MONITOR

BY: PAUL BAXTER (TRANSAM)

The TRITON Level Six Monitor is a 2K monitor program in ROM which resides on the main board, it has facilities for entry checking and running of machine code programs, in addition to many utilities. It has been written to be as accessible to the machine code, and BASIC, programmer as possible, and complements the L6.1 BASIC.

Getting it Going

The first thing to do is to remove any eproms from the main board (IC'S 21-24). The 'A' Rom should then be installed in the IC21 position and 'B' in the IC24 socket. N.B. Insert carefully as it is very easy to bend the pins upon insertion.

This done we can apply power to the system and the welcome message:

```
TRITON V6.1

END:XXXX (Where X - A HEXADECIMAL DIGIT (Ø-9 A-F)

FUNCTION? P Q J E R C G A D U H L F T S I O W M V B X N Z K
```

With the cursor at the start of the next line.

TRITON is now waiting for you to type in a character from the printed list. In the descriptions that follow reference is made to the 'main loop'. This is when the monitor has printed out the function message and is waiting for an input. The End:XXXX message will give you the address of the first location to fail (with one ram card fitted this should be 4000). This feature is executed every time a reset occurs. Note that the memory is checked non-destructively - therefore hitting reset will not destroy any programs in memory. Furthermore the memory is not initilised and will therefore have random values to start with (to initilise memory to a given value see the 'E' function).

Trying Triton

The functions and their use are described below; it is recommended that these are studied carefully if you are used to either of the previous monitors as there are some differences. After the descriptions is an example of the monitor in action - this should enable you to make full use of the features of Level Six.

P - Program. This function allows the entry and display of memory in hex - its main use is for entering and/or checking data in small blocks - for entering programs there is the 'Q' function.

To invoke this function type 'P' the message 'START:' will be printed on the next line and Triton will wait for you to type a four digit hex address (any further characters will be ignored). If you make a mistake entering the address you can change a character by using the 'DEL' key or control - H (the 'CTRL' key held down whilst 'H' is typed) and then the character(s) are re-typed. When you are satisfied that the address is correct press

the 'RETURN' key, the address just entered will be displayed - followed by the data in that location (in hexadecimal) - if the address is not valid (there are too few characters or an invalid character), the message 'ERROR' will be printed indented on the next line. If the optional oscillator is fitted this will sound for approximately ½ a second. The function message will then be printed again and the computer will re-enter the loop waiting for a letter to be typed.

When the program mode has been given a valid address it will print the address followed by the data at that location. You may now either step forward or back through memory or enter new data at that location and step forward or back. To enter data type two hex characters removing mistakes as before or if no data is to be entered step forward with a 'RETURN' or back with '↑' (SHIFT) this is used whether data has been entered or not. To leave the program mode enter control C - this is used to quit from most functions.

Q - Quick Program

This function has been included as a tool primerely for the entry of programs or strings of data and is not intended for changing single bytes. To enter the function from the main monitor loop type 'Q' - TRITON will respond with 'START:' You can then enter an address as detailed above. When the 'return' key is pressed the monitor will respond as in the program mode and you may step backwards or forwards as above. The difference comes in entering data - instead of having to press 'return' between bytes the monitor will enter and step forward as soon as two digits have been entered - in addition invalid characters (not Ø-9 or A-F) will be ignored. If a valid character is typed incorrectly you cannot use the 'Del' key - instead you must press 'return' followed by ' ' to go back to that location again. To exit from the function type control C as above. You may step forward or back through memory after entering the list digit without altering the content of that location.

J - Jump to BASIC L6.1

This function exits from the monitor to BASIC L6.1 setting up memory and vectors for the correct operation of the BASIC.
N.B. This function must be used before the 'X' function.

E - Erase Memory

This function is used to initilise a block of memory to any given value. When this function is called up the computer will request a start address, an end address and a byte value - the start and end addresses are both 4 digit hex-addresses as detailed previously, the byte value is any two digit hex byte. When 'return' has been pressed after entering the byte the monitor will proceed to fill memory.

between the two addresses (inclusive) with that byte. When the function is finished the oscillator will sound.
N.B. For correct operation the start address must be below the end address as the byte is written to memory starting at the 'START': address and incremented until it equals the end address. Therefore if the 'START:' is given at 3000 and the end as 2000 all memory will be set to the value except between 2ØØØ - 3ØØØ! This also applies to the S,H,F and N functions.

R - Register Operation

When this function is entered the monitor will wait for another key to be pressed to indicate which register operation is required. If the space bar is hit all registers and flags will be displayed - a typical printout is shown below:

```
A   C   B   E   D   L   H   SP     PC
ØØ  1Ø  FF  57  Ø5  57  AØ  1468   1647 SAP
```

you will note that the register pairs (HL, DE,BC) are displayed low order byte first - the letters refering to the register PC - Program counter, SP - Stack pointer are printed above. The letters following the program counter represent the flags - if the flag is set the letter is present. The 5 are listed below:

S = Sign Flag Z = Zero A = Auxillary Carry

C = Carry P = Parity

To modify a register from the main loop type 'R' followed by the letter of the register you wish to modify e.g. to modify the accumulator type 'RA' the value of the accumulator would then be printed - with the cursor one space along, you may then enter a new value for the accumulator followed by 'return' if you do not wish to modify the register you may exit by typing 'return' without entering any data - the monitor will signify an error condition and the register will not be changed - the same applies to the 16 bit registers SP or PC. To modify these registers type 'P' for program counter and 'S' for stack pointer. To display the flags from the main loop type 'RF' and the flags set (if any) will be displayed. If you do not wish to change the flags enter control C. To clear all the flags type return or to set some flags type in their letters (as detailed above) the letters need not be in any order. An invalid character in the string will not be flagged as an error but will simply terminate the update i.e:- PZBCA will set the parity and zero flags only, resetting the others.

TRANSAM

TRANSAM COMPONENTS LIMITED
12 Chapel Street London NW1 5DH 01-402 8137

C - Continue from a breakpoint

This function requires no other data and loads up the registers in the 8080 with the pseudo-registers in memory (those displayed in the R function) the last register to be loaded is the program counter, causing an immediate jump to the address held in PC register.

G - Goto Address

This function will execute a machine code program at a specified address - when the letter is entered a start address will be requested, once the address has been entered by pressing 'return' the monitor checks for any errors and if none are found a carriage return followed by a line feed will be printed and the program will start to execute.

A - ASCII String into Memory

This function allows you to enter a string of characters into memory directly from the keyboard. The monitor will request a start address. When this has been entered correctly and the 'return' key is pressed the monitor will print a carriage return followed by a line feed and wait for you to enter some characters - as each character is entered it is displayed and entered into memory starting at the address specified. To delete characters use the 'DEL' key or control H. When the string has been entered an 'EOT' character (Ø7H) must be entered to denote the end of the string. To obtain this character type control D - this will be entered into memory and the monitor will then print END:XXXX. Where XXXX is the address of the byte after the 'EOT' character, therefore this is the address for any more strings to be entered afterwards. If 'EOT' is not required on the end of a string, terminate it with 'control C'. The 'control C' will not be entered and the end address is that of the byte immediately after the last character.

D - Display String

This function imitates the PSTRNG monitor utility (see later) and will print out any string from the address entered until it encounters an 'EOT' character such as those entered using the 'A' function. When this function is entered the start address of the string will be requested - when this has been entered a CR/LF sequence will be printed followed by the string. Unless it is known that a string exists at the entered address, it is best to use the 'N' function to find any strings. When this function is entered the start address of the string will be requested when this has been entered a CR/LF sequence will be printed followed by the string.

U - Useful function (Port Operations)

This function allows you to read and write to any port - when this function is entered the port number will be requested - this must be entered as a two digit hex number followed by 'return', the data at that port will be displayed and you may either hit 'return' which will display the port data again (this is useful to see if the data at the port is changing) alternatively, a two digit hex number may be entered - followed by return, this data will then be written to the port and the data at the port will be read and displayed again. To exit from this function enter control C.

H - HEX Dump

This function produces a formatted hex-dump of memory between any two addresses. This allows for greater density than the 'L' function. The monitor will request a start address followed by an end address and will then produce a dump of memory (in hex) from the start address to the end address (inclusive). The dump is formatted so that the start of each line (with the exception of the first) will have the address XXXØ, therefore all lines will contain 16 bytes (except the first and last). To exit from this function before it has finished type control C. To halt the printout temporarily enter control 'S' (X-OFF) to resume printout type control 'Q' (X-ON). These actions apply to all printout with the addition that when the printout has been halted with control 'S' a new line can be started by pressing the return key, whereupon a CR/LF sequence will be printed - this is most useful when the output is diverted to a teletype or similar printer when a long line would otherwise be overtyped at the end stop - the printout can then be resumed with control Q. When the dump has finished the oscillator will sound to inform you.

L - List Memory

This function will list short sections of memory by mearly entering an address. When this function is entered a start address will be requested from the user - when this has been entered that address will be printed followed by the data in that location. This sequence will be repeated on subsequent lines for the next 15 locations and below that the prompt 'more?' will be printed. If you type 'Y' the next 15 locations will be listed, if you enter any other letter the function corresponding to that letter will be entered, if any other character is typ ed (e.g: the space bar) the error message will be printed and the monitor will enter the main loop.

F - Find Bytes

This function allows you to search memory between two addresses for any string of bytes, when this function is entered a start and end address are requested followed by the 'BYTE:' prompt. You may then enter a string of bytes separated by commas and the last byte terminated by pressing 'return'. The monitor will then start to search between the two addresses (inclusive) for that byte string and print out the locations (if any) where the string starts. When the search has finished the oscillator will sound to inform you.

T - Trap

Trap is the Triton Residant Assembly Language Package - it resides on an 8K eprom card on the motherboard (see catalogue or documentation for details). This function will enter TRAP displaying the list of options available - no memory initilisation is performed by entry to TRAP and therefore program integrity is maintained.

S - Shift Memory

This function is more than a memory move instruction it is an 'intelligent' move. When this function is entered a start and end address are requested from you. When you have entered the address of the block you wish to move (inclusive) the prompt 'TO:' will be printed - you then enter the address you wish tomove the block to. After pressing 'return' the monitor will move the block and when finished beep the oscillator to tell you.

I - Input Tape (in standard TRITON format) to memory

This function will allow you to enter pre-recorded tapes into TRITON's memory starting at 1602. When 'I' is typed from the main loop the tape control relay will be switched off and a tape header requested. When this has been entered correctly (use the DEL key where appropriate) press 'return'. The tape relay will be switched on and the message 'FILES FOUND' will be printed. Note the curser drops onto the next line. The tape will now be searched for a header corresponding to that typed in. As each header is found it is printed out and if the two do not correspond a CR/LF is printed and the monitor waits for the next header. If the monitor recognises the header the curser stays on the same line whilst the program is loaded, when the tape has finished loading, the message 'END' is printed out, the oscillator will sound and the monitor will re-enter the main loop waiting for another command. If during loading an error occurs (due to a noise spike or bad tape etc.,) a CR/LF and a question mark will be printed - note that only one question mark will be printed no matter how many errors are detected. This may also mean that the monitor may not recognise the end of the tape when it comes. If this happens you must use the Reset switch since the monitor does not read the keyboard - this also applies to the 'O' function and whilst the monitor is looking for a header or loading the tape.
N.B: It is not advisable to try to load a tape when the output is directed to a printer since if the printer may not accept characters as fast as the tape is sending (30 characters per second) the header will not be printed correctly or recognised - therefore use the 'V' function first.

O - Output a program to tape (in standard Triton format)	This function will allow you to save machine code programs starting at 1602. When you type 'O' the tape relay will switch off and the tape header will be requested. When this has been entered start the tape recorder,then hit the return key - the monitor will switch on the recorder, wait for 5-6 seconds and then proceed to write the program in the TRITON standard format (see later). When the data has been written out the monitor will wait for 5-6 seconds and then switch off the relay, beep the oscillator for a moment and say 'END' before returning to the main monitor loop.
W - Typewriter Mode	Type 'W' and the monitor will print all characters entered until a control C which aborts back to the monitor: The monitor will print onto the printer if it is selected.
M - Motor Switch	This function will toggle the tape-motor control relay - i.e., if it is off it will be turned on and vice versa. Once this has been done the monitor will print its function message and wait for another command.
V - VDU Switch	This function will direct the output to a printer as well as to the VDU, So it switches the printer output on and off. Even if you have no printer it can still be of use in slowing down the output by changing the value at locations 1402/3, the printout speed may be varied (this does not apply to V6.2P). It is initially set up to 110 Baud - for formula etc., see later. At reset the printer is off.
B - Base Conversion	This function converts from HEX to Decimal and Decimal to HEX. When 'B' is typed from the main loop the monitor will wait for a second key to be pressed. Either H or D, if neither is pressed the monitor will signal an invalid character and return to the main loop. If one of these are entered (D to convert from HEX to Decimal or H to convert to HEX) the monitor will step to the next line and wait for a number to be entered. Follow 'D' with a number from ØØØØ to FFFF (all four digits must be entered) and H with a number from Ø to 65535 (leading zeros may be omitted). Negative numbers are invalid as are numbers greater than 65535, followed by return. The number in its converted form will then be printed out. The monitor will then re-enter the main loop. N.B: It is necessary for the BASIC L6.1 to be present for this function to work - it does not, however, have to be initilised (by the J function).

X - Extended Basic	This function enters BASIC without clearing memory or initilising vectors therefore the J function should be used upon first entry to BASIC - to exit to the monitor from BASIC type control C. N.B: In BASIC to delete a character you can now use the 'DEL' key instead of control H in EDIT note the top right key (5FH) will give the delete character function.
N - Neat Dump of Strings	This function will produce a formatted text dump of memory. The monitor will request a start and end address and will produce a formatted dump between those 2 addresses (for details of format see H function). All printable upper case ascii characters (20 to 5FH) are printed as they are, otherwise a period (.) will be printed in its place - the characters are separated with spaces. When the dump has finished the tone will sound.
Z - Zap A Prom.	This function is to be used in conjunction with the Eprom programme card (available from Transam), it requests a start address and will then program the 2708 eprom with that and the next 1023 locations in memory. Two errors are possible; 'PROGRAM ERROR' - indicating that the device has a bit set to 'Zero' that should go to 'one' - this is not possible although a logic one may be changed to a zero - the device will have to be erased. The other is a 'READ ERROR' this occurs if the data programmed into the device does not correspond to that in memory after the device has been programmed. If all goes well the monitor will say 'END' and sound the tone - the tone will also be sounded in the event of an error.
K - Keyboard unshift	This function is mainly of use to those using Triton with a printer and/or the lower case graphics Rom as a letter writer or word processor. Typing 'K' will reverse the effect of the monitor's shift on the keyboard so it will act as a normal typewriter i.e. the lower case small - press shift to get upper case - this only affects the letters, numbers are asbefore. To get back to normal type Shøft-K. At reset the shift is set to normal - upper case on unshift keys.

HERE ENDETH THE LESSON..........

That concludes the description of the functions - if you've ploughed through those - congratulations!

The following section contains notes on the use of the monitors various functions as examples. This is followed by notes on use, circuits, specifications, monitor routines and addresses.

SETTING AN EXAMPLE.

In this section we shall use an example given in listing 2 to demonstrate the use of the monitor. This program inputs a character from thekeyboard and maps it on the VDU.

A few brief words of explanation are in order for those of you unfamiliar with this type of listing - it is called an assembler listing and is split into 6 columns:-

ADDRESS	DATA	LABEL	MNEMONIC	OPERAND	COMMENT
1	2	3	4	5	6

These listings are produced by the TRAP assembler. Columns 1 and 2 are those produced by the assembler, the first column listing the address, the data being from 1 - 3 bytes so one complete instruction is on one line - this makes the program much easier to read. The other columns are the original text, the 3rd column is optional and is used for referencing subroutines etc., the 4th and 5th columns are for the instruction and the 6th column contains any comments.

First of all clear memory to Ø so we can see what we are doing so type E 16ØØ, FFFF as the end (for good measure) and ØØ as the Byte, after a couple of seconds the function message will be printed after a beep - now list from 1600 to check this; type 'Y' to the 'more?' prompt, then type Q to enter the program. Type the digits checking the addresses against those listed. Once entered use control 'C', then check using the L function to list again. If any bytes are entered wrongly change those with the 'P' function. Once we have entered the program we must enter the 2 strings - the first one starts at 1622, type A 1622 (RET) ENTER A CHARACTER (CONTROL-D). The end address should be 1634 so put the 2nd string in by typing A 1634(RET) AGAIN (Y/N)?(CNTRL-D) -OK we should now be ready to run. Type G-16Ø2. If the program has been entered correctly the screen should clear and the message 'ENTER A CHARACTER' will be displayed on the second line - hit a key and the VDU will be mapped continuously, hit INT2 and the screen should settle down and the registers displayed - the PC will be somewhere between 16ØC and 161Ø and the HL register at 1000 which is the first position on the VDU - this is where the problem is - we haven't incremented it so the program will never end - we must insert an INX H instruction at 160D; but first we must make room for it - type S 160D 1642 160E so we are moving the block from 160D - 1642 up one byte - having done this we can insert (using the P function) 23 at 160D and run once again. The message will be printed with a graphic character preceeding it, this is because we moved the message by one byte, so we will have to change the address at 1604/5 - we can find where the strings have moved to by using the N function from 1610 to 164F. Put the correct addresses in 1604/5 and 1618/9 - having done this try again - the correct message is issued but we seem to have problems with the character - let's try a breakpoint - the best place to put one is at 160F. Using the P function put D7 at 160F and run again - we have now found the problem - the accumulator has been loaded with the H register so we have lost the original character, however if we save the character in the B register and map the B register instead this should solve the problem - we need to open out another gap at 1609, so using the S function move the program up and insert the byte 47 at 1609 and change the byte at 16ØD to 7Ø (MOV M,B)

having done this change 161Ø to FE - now all that remains is to change the addresses of the strings (again) so use the F function to search from 1600 to 165Ø for 45,4E and 41,47 which are the first 2 Bytes of the 1st and 2nd string respectively. Change 1604/5 and 1619/A again and run - it should now do what we said it would - and correspond to listing 2.

Now do a hex dump from 16ØØ to 165Ø to find where the program ends and put this address in 1600/2 - we are now ready to save the program. Type 'O' and give the program any tape header start the recorder and press 'return' to send the data. To retrieve the program use the I function - whilst loading the tape deliberately stop and start the tape - a question mark should be shown on the next line - indicating a load error - only one question mark will appear. This concludes the example.

NOTES ON USE:

1. When a program is run a 'ret' instruction (C9) will return to the monitor (provided the stack is not reset).
2. When using the 'F' function a match will always be found at 1410 as this is where the match pattern is held.
3. After using TRAP it is best to do a reset as some of the interupt vectors may have been destroyed..
4. The monitor checks for a program in ROM 2 (IC22) before printing out the function message. If the first byte in the second rom is 31 (LXI SP) then the program in ROM 2 will be executed.
5. If an attempt is made to jump to non-existant memory (FF) this will be vectored back to the monitor (as with all the interrupts) so vastly diminishing the chances of destroying the program.
6. Tiny BASIC will not run with this monitor.
7. An explanation of the 'S' function - if the byte sequence 31,70,1A is held in memory and it is to be moved forward (UP) one byte to give 31,31,70,1A it must be moved from the top first otherwise we get 31,31,31,31 but if we want to move the other way we must start at the bottom.

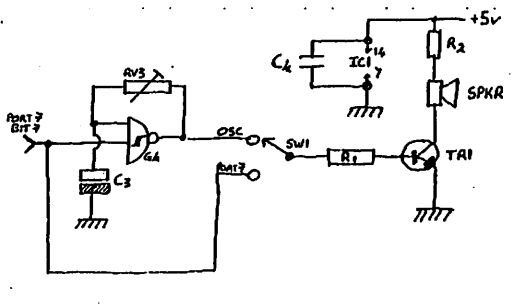
   N.B.: TOP : FFFF BOTTOM : ØØØØ UP: TOWARDS THE TOP

   DOWN: TOWARDS THE BOTTOM
8. The G function can be used for repeatedly executing a program by loading the stack pointer with 1470 and the PC with the address of the program - NOTE that in this case a 'return' to the monitor is not permissable - instead a jump to START must be used.
9. SETTING PRINTER MODE FROM PROGRAMS

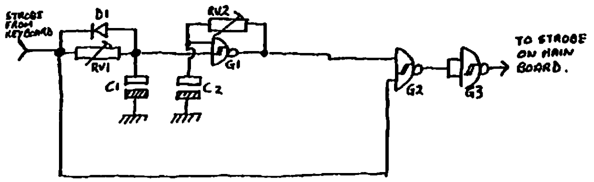
```
POKE 5120, 83;REM SET. PRINTER ON
POKE 5120,170;REM SET PRINTER OFF

3E55          MVI A,55H
32 Ø1 1A      STA 1401H      ;PRINTER ON

3EAA          MVI A,ØAAH     ;PRINTER OFF
32Ø1  1A     STA 1401 H
```

10. The Y function is vectored through ram at 1473 so to change the vector change the locations at 1474/5.

KEYBOARD_AUTO REPEAT_AND_OSCILLATOR



R1	=	1K ± 10%
R2	=	SEE TEXT
RV1,2,3	=	10k PRESET
C1	=	100µF 16V ELECT
C2	=	4.7µ 16V TANT
C3	=	1µ 16V TANT
C4	=	·1µ POLYESTER
D1	=	1N4148
TR1	=	2N3053
G1-4	=	74LS132
SPKR	=	3-16 OHMS SPEAKER.



HOW IT_WORKS_

AUTO REPEAT - with no strobe (input LOW) C1 is discharged and the input is now the output is high from G1 and C2 is fully charged. G3 o/p is high and the o/p G3 (acting as an inverter) is low. When a strobe arrives C1 starts to charge up (via RV1) but whilst it is changing (time determined by setting of RV1) the o/p of G1 is still high, since both inputs to G are high the output goes low - this is inverted by G3 sending a steady strobe to the input port - if now the strobe input goes low again C1 discharges rapidly via D1 and the output goes low again, if, however, the input is maintained for long enough for C1 to charge up, both inputs from G1 are high and the output goes low - this sends the output (by G2+3) low again, this starts to discharge C2 - when C2 has discharged the input goes low sending the output high and giving a high output again - this starts to charge C2 again and so on until the strobe is removed. ~~P~~RV2 determines the rate of repeat.

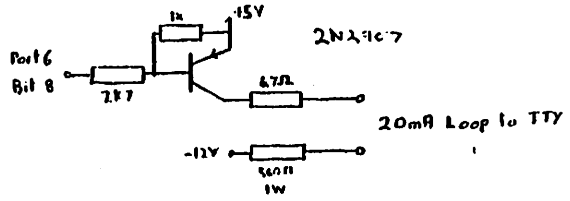
OSCILLATOR_

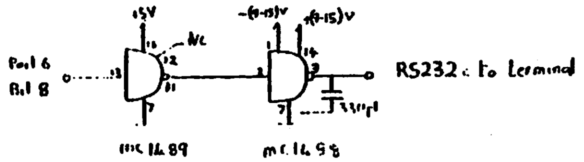
The principle of operation is similar to that of the repeat, but when the input (BIT 7) is low the output of the gate is held high, but as soon as the output goes high the circuit starts to oscillate. Since C3 is small this occurs rapidly giving an audio frequency (adjustable by means of RV3), this is amplified by TRI to drive a small speaker. R2 should be chosen to give the appropriate tone - any value from 20R to 100R should be alright

Alternatively, a 30R resistor may be placed in series with a 100R wire around potentiometer to act as a volume control. The switch included is optional but is desirable to allow the speaker to be switched direct from port 7(for music programs etc), the oscillator should sound on an error and on the other functions detailed previously.

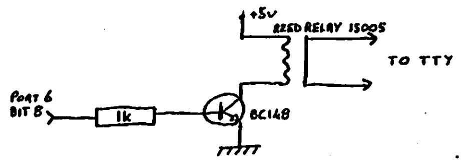
CONSTRUCTION_

The circuit is reasonably simple and should pose no construction problems. Veroboard is one suitable method, its prototype was built on vero-strip board and mounted between the switches and transformer next to the speaker. The components should be mounted resistors first - the IC should be socketed - put C4 as close to IC1 as possible - this will stop weird effects when the oscillator sounds with outo repeat. Take care to insert the capacitors and diodes in correctly, if the diodes is reversed the repeat will start immediately.





The two circuits shown above are to interface the serial output from Triton to a 20ma or RS232 terminal - on certain TELETYPES* making the circuit below should suffice where the TTY supplies its own 20ma

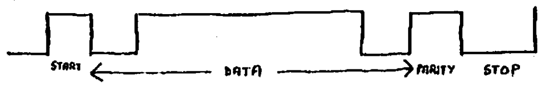


The reed relay is the same type as used on the tape I/O on Triton.

*Trade Mark of TELETYPE CORP.

SERIAL_I/O_

The level six monitor provides a serial output to a printer. This can be turned on or off by using the V function, it can also be changed in programs as mentioned previously. The serial output appears as one bit on Bit 8 of Port 6 it is overided in software and the speed of output can be easily changed by altering a value stored in RAM - see later.
The data is inverted - i.e. logic Zero is +5 and one OV. The format is as follows:-



Each character starts with a start Bit followed by 8 data bits followed by a fake parity Bit (always 1) this is suitable for teletypes and most printers where parity checking is not needed. Two stop bits are sent at the end of each character with the exception of a carriage return where a continuous stop bit is sent for 3 character periods to allow the mechanism to return properly, this should be more than adequate for most machines. N.B. A special version of the monitor is available for high speed parellel printers such as the TRANSAM BD80. It should be noted that the character is output to the printer (if selected) before the VDU.

When the computer is reset the serial o/p speed is set to 110 baud - Bit time = 9.09ms = 10 characters per second, to alter the speed the formula below should be used to calculate the value - converted to hex, and put into locations 1402/3.

$$\text{Tbit} = \frac{1}{\text{Baud rate}} \text{ sec}$$

$$\text{tcy} = \frac{9}{\text{clock rate}} \text{ sec}$$

$$\text{tcy} = 1.255\text{uS for } 7.168\text{MHz clock}$$

$$\text{oR } 0.5\text{uS } \text{"} \ 18\text{MHz } \text{"}$$

$$n = \frac{\frac{\text{Tbit} - 107}{\text{tcy}}}{24}$$

eg for 110 Baud

$$\text{Tbit} = \frac{1}{110} = 9.09\text{mS}$$

$$n = \frac{\frac{9.09\text{mS}}{1.26\text{uS}} - 107}{24}$$

$$= 297 \quad \text{or } 0129\text{H}$$

The minimum value (0001) gives 6K baud at 7mHZ or 15K baud, at 18mHZ.

TAPE_FORMAT

The standard Triton format is used for input and output in the 6.1 monitor - this is detailed below.

Each Bit consists of a start Bit, 8 Data Bits, an odd parity Bit followed by 2 stop Bits. The actual record format is 5-6 seconds of continuous mark tone, 64 sync characters (ODH), program end address low, high, program, 5-6 seconds mark tone. The end address is stored in 16ØØ,Ø1, this is automatically set up by tiny BASIC but must be preset by the user in machine code programs (see example) for BASIC L6.1 the tape routines are contained within the BASIC and these should be used for loading and storing L6.1 BASIC programs.

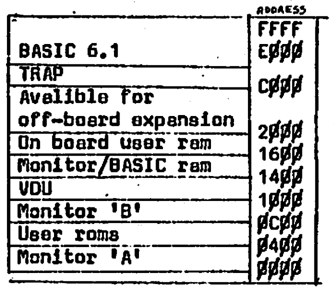
MONITOR ROUTINES_AND_ADDRESSES_

MEMORY MAP:

This is a memory map of the L6.1 system.

The L6.1 monitor occupies locations ØØØØ to Ø3FF

and ØCØØ to ØFFF, the space from Ø4ØØ to ØBFF is left for user expansion (see previous section).



Useful addresses are listed below.

1401	DISSW	55 = PRINTER ON	AA = VDU ONLY
1403	SPEED	Baud rate	
1410	BUFFER		
143Ø	INT 3		
1433	INT4		
1436	INT5		
1439	INT6		
143C	INT7		
1472	KEYSFT	8Ø = NORMAL	7F = TYP$\overset{E}{\wedge}$WRITER SHIFT
1476	INVEC	INPUT VECTOR	
1479	OUTVEC	OUTPUT VECTOR	

PORT ADDRESS

Ø -	Keyboard Input only special routine needed
1 -	Tape I/O Uart status input
2 -	Tape I/O Uart data strobe output
3 -	Led port output On = Logic ZERO
4 -	Tape I/O uart recieve data enable input
5 -	VDU output Strobe routine needed
6 -	Serial O/P on Bit 8 Bit 7 Spare
7 -	Bit 8 = Relay Bit 7 = Oscillator (speaker) (see circuit)
8 - 15	Decoded on baord by LS 154 (needs extra decoding)
16- FF	Available for extensions

MONITOR UTILITIES

ADDRESS	NAME	COMMENTS
ØØØØ	RSTØ	Reset address. Sets up vectors - checks memory and prints initilisation message after clearing screen.
ØØØ8	RST1	Clears VDU and resets curser returns after suitable delay with all registers intact.

ADDRESS	NAME	COMMENTS
ØØØB	INCH	A call routine which waits for a character to be typed in from the input device and returns with the character in the accumulator all other registers intact.
ØØ1Ø	RST2	Interrupt 2 - Jumps to INIT routine saves registers and prints them on VDU and printer (if sel, ected) then enters main loop.
ØØ13	OUTCH	Character output routine;outputs to the VDU (and printer if selected) the character in the accumulator,returns with all registers intact.
ØØ18	RST3	INTERUPT 3;vectored to 143Ø for user jump vector. (NOTE: on parellel printer versions this interrupt is not available to the user).
ØØ1B	INDATA	Will wait for characters to be entered from input device,echo on the VDU and put into memory pointed to by the DE register pair - on a carriage return the routine terminates;a space followed by an EOT character (04H) is stored in memory with the DE register pointing to the EOT character, the return is not stored.
ØØ2Ø	RST4	= Interrupt 4;vectored to 1433 for user operations.
ØØ23	PDATA	Prints a string stored in memory pointed to by the DE register pair;returns on EOT with all registers intact with the exception of the DE pair, points to character After the EOT - allowing the routine to be called again for a string following the first.
ØØ28	RST5	= Interrupt 5;vectored to 1436 for user operations.
ØØ2B	PSTRNG	Similar to Pdata but a CR/LF sequence is output before the string is printed.
ØØ3Ø	RST 6	= Interrupt 6;vectored to 1439 for user operations.
ØØ33	PCRLF	Will output a CR/LF sequence to the output device(s) returns all registers intact.
ØØ38	RST 7	= Interrupt 7;vectored to 143C for user operations.
ØØ3B	ECHOCH	Will input a character from the input device and print it on the output device before returning it in the accumulator - all other registers intact.

The above subroutines are fixed utilities and are quaranteed to remain at these addresses if another monitor is produced.

The routines listed below are not guarnateed to remain in these positions if subsequent versions of the monitor are released.

Routines marked with an asterisk (*) are in the same position as the V5.1 monitor.

*URTOUT	ØØ3E	Will output the character in the accumulator to tape, returns registers intact.
*TAPOFF	ØØ4A	Will switch the tape motor relay off - returns with A=Ø all other registers intact.
COMP	ØØBF	Will compare the DE to the HL register pair,returns C set if DE HL Z set if DE=HL and neither set if HL DE. A is lost.
*FIVSEC	ØØC5	Waits for 5-6 seconds before returning all registers intact.
RESPRP	ØØE5	Will print the CR/LF sequence followed by the string pointed to by the DE register pair and will then wait for a character to be entered, echo it and return with the character in A, the DE register pair will point to the location of the EOT character - all other registers intact.
ERROR	Ø15B	Will print a CR/LF followed by the message 'Error' and beep the oscillator before returning to the main loop.
ERR1	Ø15E	As above but without the leading CR/LF
PRBEEP	Ø15A1	Will print the string pointed to by the DE register pair,beep the oscillator and return to the main loop.
BEEP	Ø154	Will sound the oscillator and return to the main loop.
START	Ø174	Re-enters the monitor main loop.
STRTAD	Ø2Ø8	Will print the prompt 'START:' and get a 4 digit address in the HL pair DE + A lost - aborts to monitor an error.
GETADR	Ø2ØB	Will print the message pointed to by the DE pair and get an address as above.
G2BYT	Ø2ØE	Will print a space and wait for an address to be entered as above.
STEND	Ø22A	Will get a start and end address, end on top of stack - start below that. HL,DE,+A registers lost.
COMP I	Ø231	Similar to COMP but jumps to beep if DE-HL are equal.
PRADAT	Ø23F	Prints the address pointed to by the HL pair and the data in that location on the next line. A is lost, others intact.

PRTLOC	Ø242	Prints the data pointed to by the HL pair - A lost.
ENDADR	Ø251	Prints the message 'END:' followed by the address in the HL pair - A and DE lost.
PRAD~~Ø~~1	Ø254	Prints the address in the HL pair.
PHEXSP	Ø259	Prints the data in the accumulator followed by a space. A lost.
PRTSPC	0256	Prints a space A - lost.
INSPBF	Ø27B	Prints a space then gets a string in the Buffer returns on 'return' with a space followed by an EOT. DE points to EOT character - DE + A lost.
INBUFF	Ø270	Prints the character in A and gets a string in buffer as above - DE + A lost.
PRTDAT	Ø2AB	Prints the accumulator in hex - A lost.
TAPOUT	Ø2C2	Turns the tape relay off, requests a header - gets header, turns relay on, waits 5 seconds, sends 64 sync characters followed by header and returns. A,B,D,E lost
TAPIN	Ø2E1	Turns the tape relay off, requests and gets a header - turns the tape relay on prints the message 'FILES FOUND:' and searches the tape, printing all headers returns when a match is found. A,B,D,E lost.
URTER	Ø365	Gets a character from tape and checks for an error condition, prints out a CR/LF followed by '?' N.B. The C register must be loaded with '?' prior to calling this for the first time but not changed until tape loading is complete - B lost.
TAPRD	Ø3E5	Gets a character from the tape and prints it on VDU - repeats indefinately.
*TAPON	Ø3F6	Turns tape relay on. A returns 8ØH All others intact.
GETACC	ØC14	Expects to find 2 hex digits in buffer converts and puts in A.- B + DE lost.
*UARTIN	0E62	Gets character from tape in A - other registers intact.
*TAPHDR		Turns relay off, requests and gets a header in buffer. DE + A lost.
ERRMSG	ØED6	Start address of string which prints 'ERROR', call at ØED7 to omit space. (use via PDATA OR PSTRNG).
STRMSG	ØECF	Start address of string which prints 'START:' Use as above.
ACKA	ØFDS	Start address of string which prints 'HEADER:' Use as above.

ACKD	ØE49	Start address of string which prints 'END' Use as above.

Notes for use with programs in TRITON manual.

1. Using interrupt 3.

   Interrupt 3 is now vectored to 1430 so you should first insert thebytes C3,18,16 in locations 1430-2 before using the program.
2. Duo-Decimal program.

   Change the instruction at 1615 to RET by changing 1615 to C9.
3. Keyboard/Led Program version 2.

   Change the instruction at 160E in the alternative ending to RZ by changing the byte at 1610 to C8.
4. Alphabet twelve times using I/O - change instruction at 1615 as above.
5. Alphabet using memory mapping.

   change the instruction at 1626 as in 3.
6. Modem echo test.

   Change the instruction at 1602 to call urtout by changing the bytes at 1603/4 to 3E and ØØ - change the instructions at 1606 to call uartin by changing the bytes at 1607-8 to 62 and ØEH.
7. Test tape playback program.

   Is now contained in the monitor at Ø3E6.
8. Test tape record program

   Change the instructions at 1602, 160D, 1612 to call urtout by changing the bytes following each of these addresses to 3E and ØØ.

IMPLEMENTATION

Level Six is available direct from TRANSAM COMPONENTS LIMITED at 12 Chapel Street, London, N.W.1. They can also be supplied in exchange for programmed eproms (see catalogue for details). The L6.1 BASIC and TRAP are also available from the above address.

TRANSAM

TRANSAM COMPONENTS LIMITED

12 Chapel Street London NW1 5DH 01-402 8137