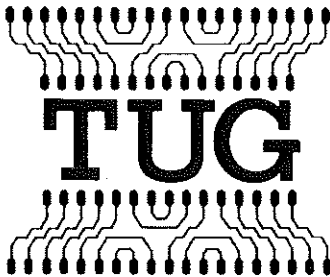


TANGERINE USERS GROUP



NEWSLETTER

ISSUE 10

EPROM PROGRAMMER

We are now launching a simple eprom programmer for the 2716 and its equivalents. The programmer has been designed to be a cheap alternative to those already on the market. We are providing the single sided p.c.b., construction notes, instructions on use and programming tips, along with powerful software to enable you to programme your eproms direct from memory contents automatically. The programmer is powered by three 9v PP3's (for 27v) and is connected to the three ports A1, B1 & C1 on Tanex (parallel mode). Six commands allow you complete control over programming: BLANK EPROM/PROGRAMME EPROM/LIST CONTENTS/VERIFICATION OF CONTENTS/TEST CONTENTS/SCREEN MONITORING DURING PROGRAMMING/ABORT.

Components will not be provided as we assume that many of the parts will be found in the constructor's spares box, any others being obtained cheaply at any major retailer. Apart from the connecting leads, there are seven components required on the board, which we feel should be easy enough to construct for any member.

EPROM PROGRAMMER KIT (less parts) £17.00 including VAT + p & p. (TWO 6522 VIA's are required).

Please keep up the good work on the newsletter; I find it invaluable, especially the programs. Long or short they provide good brain exercise following them through. As I am in a hardware related occupation, I find few problems with the "nut & bolts", but stumble with the software. However, I'm working on it.

The monitor problem (A.W. Gwent): If as I read the letter, he is connecting the interface to the High impedance (Hi Z) input, I feel that this is the cause of the trouble, i.e. the Low Z output from the BC108 is mis-matching the High Z input of the monitor. He should redesign the interface to give a higher impedance output and screen off the Microtan and interface. I had similar vertical dotted lines etc. on my TV using the standard modulator arrangement, which stopped when I put the Microtan, Tanex and Keyboard in an aluminium case. If A.W. Gwent cannot design an interface, if he could contact me, I will see if I can help.

Philip Blundell, 38, Linden Road, Barton under Needwood, Burton on Trent. DE13 8LN. Tel: Barton under Needwood 2123.

V2 REVIEW

Tanbug V2 is the new monitor that replaces the 2K eprom Tanbug on the Microtan 65 board. Whilst it is not yet on general release (awaiting printing of the manual), a few have been issued to those of us who can't wait to get their printers running. Here are some of the additional features that V2 has to offer:

Full printer driving software: using Tanex ports C1 and D1 (for a parallel printer), or port E1 (for a serial printer), any character that is printed on the screen can also be printed by the printer, with full printer fault checking. This fault checking monitors the state of certain port signals, so if a serial printer is not fitted then pins 8, 9 and 11 on E1 have to be tied to +5v (pin 14) through a 1k resistor (a header plug is supplied with the chip). The printer routine is turned on and off by typing CTRL P/S. Doubtless this can also be achieved under software control, but without the manual I haven't found out how. The computer can also be made to respond to a teletype with keyboard, attached to the serial port.

Basic entry: Basic users now need only type "BAS (return)" to enter Basic, and if there's been a accidental reset the "WARM (return)" gives you a Luke Warm start. This won't work if you've reset during an X-BUG tape routine, or if Zero page has been corrupted.

Screen clear: when in Basic or X-BUG, typing CTRL-L or printing CHR\$(12) clears the screen (filling it with CHR\$(32)), and the cursor reappears on the top line. Subsequent "return"s step it down one line without the screen scrolling, until it reaches the bottom line, where the screen behaves normally. At last you can use the print routine and have mixed characters and graphics. (A useful little routine for graphic clear screen in Basic is NULL 255 : ? : ? : NULL 0 which fills nearly all of the screen with character zero). The routine does not alter \$CA - \$OB (cursor line address), so you can't steer the cursor around by altering that value. In fact, techniques to keep the bottom line unscrolled don't work with V2. The only other thing to watch out for is the X-BUG cassette software, which examines the screen rather than using the character input routines (which remain unaffected). This may mean that although you want to examine program "INVADE.A" at fast speed you are in fact loading program "A" at slow speed. After a while, pressing "repeat+return" before typing "SAVE" or "LOAD" becomes automatic. If you haven't used CTRL-L recently then of course it's not necessary. This method of printing results in overwriting when exiting from Basic edit mode - occasionally an additional space is inserted into the line being edited. Again this only happens if the cursor is not at the bottom of the screen, and you soon get used to ensuring that this doesn't happen.

Apart from the trick of an unscrolled line, all programs written using Tanbug V1 are totally compatible - the problem of the cursor not being on the bottom line doesn't arise since this doesn't occur in V1 programs. For instance, TUG's Tanscribe (Text Editor) works with the printer, but don't store lines over 80 characters because you don't get auto-return. Similarly, most programs written for V2 that don't use printers will run happily on V1, but watch out for replacing "PRINT CHR\$(12)" with "FOR I = 1 TO 16 : PRINT : NEXT" - if the program uses screen locations or mixed graphics and alphanumerics then it may not behave as originally intended.

These features aren't all that's behind the new Tanbug - CTRL-V has some effect, and CTRL-Q is only echoed on the screen with alternate keyings, so something's going on there as well, but without the manual . . . who knows known? (Watch this space!)

The appearance of Tanbug V2 does represent another step forward for Tangerine, for now a Tangerine system can form a minimum business system for someone who doesn't know what makes a computer tick. It may not have disks and it still needs software and manuals; but now that printer driving firmware is available, the latter may not be too long in arriving. Even without the promised hi-res colour graphics board a simple word processor for all those letters to TUG is now possible. So, let's hear from all those budding software writers!

B. Walker 2, Hill Road, Theydon Bois, Epping, Essex. CM16 7LX

Dear Bob,

With reference to the comment in issue 9 of the newsletter, I did in fact use the $\overline{I/O}$ signal in my original design. After spending three nights with a "scope", I decided to cut my losses and fully decode all the address lines. It is possible that the problem is inherent in my machine, but when address BCO0 was being accessed, the $\overline{I/O}$ signal was not going low. I thought I should write and explain my reason for not using it. (The chip only costs about 20 pence).

Andrew Brown 5 Oswald Road, Ayr, Scotland. KA8 8NY

Dear Bob,

I enclose a short routine which is useful in conjunction with the disassembler. If one wishes to disassemble a long program to look at one particular bit, it is useful to be able to stop the disassembler, then make it continue from where it stopped, without having to press reset (assuming you used the line feed command). Using the CR command is tedious.

If you enter the following program, from LFC0, type G1FC0, then, as long as reset is not pressed, from then on, when using the disassembler, press linefeed and the disassembler will pause whenever control S is pressed, and continue when control Q is pressed. Normally, having used the CF command, the only way to stop the disassembler is reset, which is inconvenient, especially if you have a clock routine running. However, control S followed by ESC will now return you to the monitor.

This routine unfortunately does not work for the Basic 'List' command, which would be useful. (I think it does not work because Basic disables interrupts during listing). If anyone comes up with a Basic compatible version, I would be very interested.

```
1FC0 LDA #4C      - SET UP INT S1
1FC2 STA #10     TO INT S3
1FC4 LDA #D2
1FC6 STA #11
1FC8 LDA #1F
1FCA STA #12
1FCC JSR #FE73
1FCF JMP #FC37   - RETURN TO MONITOR
1FD2 PHA        - SAVE A
1FD3 LDA #1
1FD5 CMP #13     - CTRL S PRESSED ?
1FD7 BEQ #1FDB   - YES
1FD9 PLA        - NO, RETURN
1FDA RTI
1FDB TYA        - SAVE X,Y
1FDC PHA
1FDD TXA
1FDE PHA
1FDF CLI        - ENABLE INTERRUPTS
1FE0 JSR #FDFA   - GET CLEAR
1FE3 LDA #1
1FE5 CMP #11     - IS IT CTRL Q ?
1FE7 BNE #1FF0   - NO
1FE9 PLA        - YES, CONTINUE
1FEA TAX
1FEK PLA
1FEL TAY
1FEO JMP #1FD9
1FF0 CMP #1B     - IS IT ESC ?
1FF2 BNE #1FE0   - NO, LOOP
1FF4 PLA        - YES, RESET JUMPER
1FF5 PLA
1FF6 PLA
1FF7 JSR #FE73   - RETURN TO MONITOR
1FFA JMP #FC37
```

Perhaps you would be interested in publishing the enclosed in the newsletter. P. K. in issue 2 asks if one can save/load specific lines of a program or merge 2 files. I enclose a method of merging that I have found; saving specific lines is probably possible (I'm working on it!), but loading specific lines can only be done by loading the whole file and using a "merging" method with the new file, like the one I enclose.

PROCEDURE FOR FILE MERGING

To understand how the procedure works, it is necessary to understand how BASIC stores programs.

Example—the line 4567 REM ABCD is stored as follows---

```

0400  00
0401  0C01  Pointer to start of next line
0403  0711  Binary code of linenumber-4567
0405  0E     Single byte code for REM
0406  20     Space
0407  41424344 ASCII for ABCD
0408  00     "Break" indicating end of line.
040E  XXXX   Start of next line as pointed to at 0401

```

(If 4567 was the only line, then "XXXX" would be "0000" indicating end of program)

My method of merging to files, is to first arrange the linenumbers of the two files so that all of one file's linenumbers are less than the other's. I then shift the latter file up in BASIC RAM above the point where the first file would have finished. One can then join the two together.

The following BASIC ZERDPAGE locations are required---

```

9A/9B  Start of BASIC RAM, normally 0401 (NOT USED)
9C/9D  First free location after end of program
9E/9F  ) First free location after variable store
A0/A1  )

```

(Variables are stored immediately after the program except for the string data which is at the top of available memory.)

Since , for this purpose , no variables are in use,
9C=9E=A0 A0 9D=9F=A1

Procedure.....

- 1) Ensure file A's highest linenumber is at least 100 less than file B's lowest linenumber.
- 2) LOAD file A, and add a line at the END which is easily recognisable—I suggest REM***** -referred to in future as RECA.
- 3) Type PRINT PEEK(157)+256+PEEK(156), and note the value. (Referred to in future as X). SAVE file A.
- 4) LOAD file B, and add a line at the BEGINNING which is easily recognisable - referred to in future as RECB.
- 5) Type PRINT PEEK(157)+256+PEEK(156), note the value. (Referred to as Y)
- 6) Using linenumbers between those of files A and B, add to file B sufficient lines of rubbish so that when PRINT PEEK(157)...etc. is typed, the new value is 2 or 3 more than X+Y-1024.
- 7) Do not SAVE, just LOAD file A. RETURN TO TANBUG.
USING ONLY L AND M COMMANDS-

- 8) List from 0400 until you find RECA followed by a little rubbish and RECB.
- 9) Add to RECA, in my case using *,code 2A, up to but not including the "Break" code 00 preceding RECB
- 10) Alter the Pointer of line RECA to the start of RECB, so that BASIC "thinks" that the two files are one, so to speak.
- 11) List, through the program until you find the end of file B. There should be 3 final 00's. Store the first address following these 3 00's in locations 9C39D, 9E39F, A03A1.
- 12) This is to tell the BASIC the new end of the program. After the "final 3 00's" in para.11), store 3 more 00's.
- 13) Finally, Re-enter BASIC using a UARM start, and list. The files should be merged. Delete RECA and RECB and SAVE new file.

```

0400 20 FA FD JSR FOLLRS
0403 A5 01 LDA 01
0405 E9 30 SBC #30
0407 0A ASL A
0408 0A ASL A
0409 0A ASL A
040A 0A ASL A
040B 85 50 STA 50
040D 26 FA FD JSR FOLLRD
0410 A5 01 LDA 01
0412 E9 40 SBC #40
0414 10 00 BPL 0423
0416 A5 01 LDA 01
0418 E9 2F SBC #2F
041A 18 CLC
041B 65 50 ADC 50
041D 20 75 FE JSR QPCHR
0420 4C 00 04 JMP 0400
0423 18 CLC
0424 69 09 ADC 09
0426 4C 1A 04 JMP 041A

```

This program allows typing with a hexpad - just enter the ASCII code for the character required.

Bill Craswell

```

5 FOR CLR = 1 TO 15: PRINT : NEXT CLR
10 E$ = "ANYONE FOR TENNIS ?"
20 A = 1: B = 30: C = 0
30 A$ = RIGHT$(B$,A) + LEFT$(B$,B)
40 B$ = A$
50 GOSUB 300
60 C = C + 1
70 IF C = 12 THEN D = A: A = B: B = D: C = 0
80 GOTO 30

300 POKE 3,0: POKE 48,0
310 PRINT A$
320 FOR DELAY = 0 TO 50: NEXT DELAY
330 RETURN

```

(Illustrating use of subroutine to print on bottom line repeatedly without scrolling)

Bill Craswell

First an answer to David Combs. It is the machine and nothing to worry about! All computers which work with floating point numbers have a limited degree of accuracy. Ours is better than most mainframes where about 7 significant digits seems to be the norm. The computer has to convert our decimal numbers into binary and it may not be able to find an exact representation for what we give it. It chooses the nearest. For 365.22 it may happen that the nearest binary representation has a value of 365.220000007; for 365.11 it might be 365.109999993. Subtract them and the result is as published. It looks messy, and if we don't like it the answer is to restrict the number of decimal places. Try the following:

```

10 A = 365.22
20 B = 365.11
30 R = A - B
40 GOSUB 900
50 PRINT R

900 R = R * 1000 + 0.5
910 R = INT (R) / 1000
920 RETURN

```

The addition of 0.5 is to ensure that R is not rounded down by INT if the computer's value is slightly less than the true value. Using 1000 gives three decimal places. A more general solution would be to use 10^N where N is the number of decimal places required. Approximation error gets worse as the result gets smaller relative to the numbers you are working with. The result of a calculation such as $9999.9999 - 9999.9998$ would be unpredictable. The answer to David Combs' other problem is identical: $1 - 1$ comes out correct because it has an exact representation in binary. 0.1 does not so when you negate it and add it to itself, as approximation errors come into play.

Turnin back to page 1, the same explanation accounts for $20 * 20$ and 20^2 . Two different processes are use with different approximation errors. Try $1 * 1$ or $16 * 16$ and it will probably be the same either way, as these numbers have exact binary representations. Again, if you don't like the result, restrict the number of decimal places before output, but don't blame the computer -- it does it's best within the limits of a finite RAM!

Incidentally, for simple powers like squares and cubes, $X * X$ is preferable to X^2 . It runs much more quickly. Page 5: I don't think it is strictly true that the INPUT or PRINT statement must immediately follow the POKE, though it makes for clear, unambiguous programming if the rule is followed. The exception I would make is where the output required is conditional upon the state of some variable. As a rather trivial example, here is an extract from my file handling program:

```

POKE 22,254
IF T$(I) <> "*****" THEN PRINT CHR$(34); T$(I); CHR$(34)
IF T$(I) = "*****" THEN PRINT T$(I)

```

It works. In this particular case I think the condition is unnecessary. I inserted it during program development because I was failing to find my "end of file" marker. I later discovered that I had erased it elsewhere in the program!

Page 4: The start address of the binary - decimal converter is worth a year's sub to TUG! Renewal enclosed!

E. J. Jewell 4, Hawthorne Road, Stapleford, Cambridge, CB2 5DU

May I repeat my earlier request for some possible guidance in the matter of matching an MX80 printer to the Tangerine system. In the parallel mode and using Tanbug II printer routine, I cannot get the TAB(X) instructions to respond. I have tried the parallel routine in previous Newsletter(s) but cannot get the printer to tab in the correct position. Sometimes, due to this it will omit some of the print message i.e. "Testing" would appear as "ting" and not in the position shown on the monitor. If the following instruction is given:

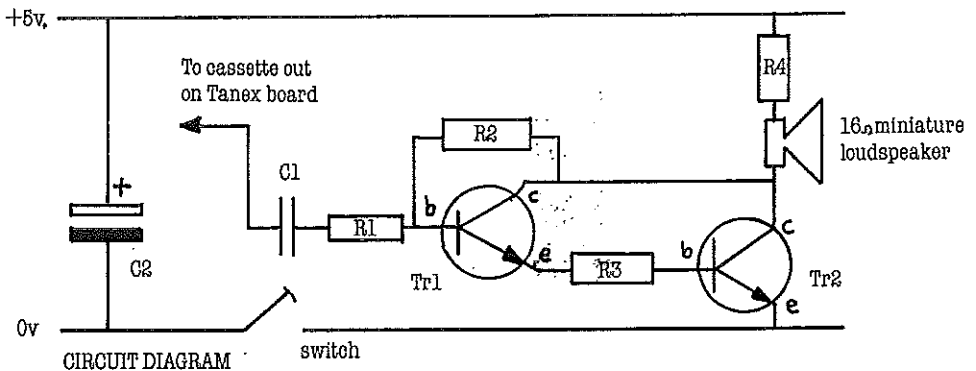
IO ?TAB(10)"M"

The monitor will place an 'M' in TAB(10) position, but the printer will not respond. I have since tried a serial RS 232 interface with a serial routine but with little success. Perhaps someone in the Group has successfully linked these two units?

Yours faithfully, L. G. Phillips

☆☆

A Simple Sound Generator



COMPONENTS:

Switch

Capacitors: C1 - 0.1 F, C2 - 450 F electrolytic

Resistors: R1 - 1k Ω , R2 - 1.5 M Ω , R3 - 5.6k Ω , all 1/4 watt. R4 - 50 Ω to 100 Ω 1/2 watt, depending on the volume required (I used 50 Ω). 30 Ω with a potentiometer to vary the sound level could be used.

Transistors: Tr1 - BC109 or BC107, Tr2 - BD131 (3 amp transistor) or any other n-p-n 1 amp transistor.

Tr2 is not a BC109, as this would mean it is working flat out continuously, and will fail, so a slightly more powerful transistor is used. Typical consumption of amplifier 100 - 200 mA.

Continued over

Sound generator cont.

C1 is present to any direct current from entering, only allowin alternating current to pass. This is connected to "cassette out" on Tanex board (which in turn goes to PB7 via 22k Ω resistor. So, if you haven't enough room to put this in your computer box, then it can easily be connected via the cassetts record socket.

I used a 16 Ω small speaker, but an 8 Ω will do. C2 is an electrolytic capacitor, placed outside the switch so that constant switching on and off does not cause excess current to flow, and also prevents any mains pickup on the power supply leads to the circuit. The lead to C1 should be screened to prevent any mains pickup. R3 is there to reduce the gain of the darlington pair transistors. Without this the circuit (when switched on but not producing any sound) made a 100Hz hum in the loudspeaker, from a 100Hz full wave rectified but not very smoothed current in the power supply. As the output from the Tanex board is via a 22k Ω resistor, R2 was needed to provide some feedback for the circuit to work.

The above circuit is only an amplifier. So you could disregard the above circuit and connect the record socket from your computer straight to your hi-fi system amplifier input.

SUBROUTINE TO USE SOUND GENERATOR :
(Relocatable anywhere)

Subroutine start

```

0050 A900 LDA #0000    DISABLE INTERRUPTS
0052 80CEBF STA #BFCE
0055 A980 LDA #0080    ENABLE PB7 AS OUTPUT
0057 80C2EF STA #BFC2
005A A9C0 LDA #00C0    SET TIMER 1 TO CONTINUOUS
005C 80CBEF STA #BFBC  COUNTDOWN WITH SQUAREWAVE ON PB7
005F AD4000 LDA #0040  LOAD TIMER 1 LATCH LOW
0062 80C6BF STA #BFC6
0065 AD4100 LDA #0041  LOAD TIMER 1 LATCH HIGH
0068 80C76F STA #BFC7
006E 80C5BF STA #BFC5  INITIATE TIMER 1
006E AD0B LDY #000B    DELAY
0070 LOOPA2FF LDX #00FF
0072 LOOPCA DEX
0073 D0FD BNE LOOP
0075 89 DEY
0076 D0FB BNE LOOP
0078 A900 LDA #0000    STOP SOUND
007A 80CBEF STA #BFCE
007D 80C2EF STA #BFC2
0080 60 RTS

```

START - NOT RELOCATABLE

```

0088 A900 LDA #0000    SET UP FREQUENCY
008A 8541 STA #0041
008C A9FF LDA #00FF
008E 8540 STA #0040
0090 20FADF POLLKB
0092 205000 JSR SUBROUTINE
0094 4C9000 JMP 0090

```

This program gives you a note every time the keypad is pressed. If you want a continuous note omit lines 78 to 98 and put 00 in memory location 78, in which case start from 0058. Before going into the subroutine save index X, index Y, accumulator and status word if needed; memory locations 40 & 41 should be loaded to produce the required frequency!

```

0040 BA
0041 DC

```

frequency produced = f

$$f = \frac{2 \times ((D \times 16^3) + (C \times 16^2) + (B \times 16) + A)}{750,000}$$

```

0050 A900 LDA #0000 DISABLE INTERRUPTS
0052 80CEEF STA #BFCE
0055 A980 CDA #0080 ENABLE PB7 AS OUTPUT
0057 80C2EF STA #BFC2
005A A9C0 LDA #00C0 ENABLE T1 AS CONTINUOUS
005C 80C8EF STA #BFCB COUNTDOWN OUTPUT ON PB7
005F A540 LDA #0040 LOAD T1 LATCH LOW
0061 80C6EF STA #BFC6
0064 A541 LDA #0041 LOAD T1 LATCH HIGH
0066 80C7EF STA #BFC7
0069 80C5EF STA #BFC5 INITIATE NOISE
006C A001 LDY #0001 DELAY
006E LOOPA2F0 LDX #00F0 VARIABLES
0070 LOOPCA DEX
0071 D0FD BNE #0070
0073 88 DEY
0074 D0F8 BNE #006E
0076 60 RTS

START

0077 A910 LDA #0010 40 & 41 CONTAIN THE INFORMATION
0079 8541 STA #0041 FOR TIMER ONE'S LATCHES LOW
007B A9FF LDA #00FF AND HIGH RESPECTIVELY i.e.
007D 8540 STA #0040 COUNTDOWN FOR HALF WAVE CYCLE.
007F 205000 JSR #0050 ; INITIATE NOISE
0082 A540 LDA #0040 ; INCREASE FREQUENCY PRODUCED
0084 38 SEC
0085 E901 SBC #0001
0087 8540 STA #0040
0089 800F ECS #009A
008B A541 LDA #0041
008D 38 SEC
008E E901 SBC #0001
0090 8541 STA #0041
0092 C901 CMP #0001 ; IF FREQUENCY REACHED A MAXIMUM
0094 D009 BNE #009A ; START LOW FREQUENCY AGAIN
0096 A903 LDA #0003
0098 8541 STA #0041
009A CDTA540 LDA #0040 ; TO CHANGE FREQUENCY JUST LOAD
009C 80C6EF STA #BFC6 ; NEW TIME INTERVAL IN TIMER
009F A541 LDA #0041 ; ONE'S LATCHES
00A1 80C7EF STA #BFC7
00A4 206C00 JSR #006C ; DELAY
00A7 4CB200 JMP #0082 ; CONTINUES TO DECREASE FREQUENCY

```

To change this to a laser program, alter memory location 6F to 64 and restart program from 0077 (this decreases delay).

To make missile firing program (press any key to fire) get siren program and alter the following :-

Memory location 4F to 01

```

0077 A700 LDA #0000 INITIAL FREQUENCY HIGH
0079 8540 STA #0040
007B 8541 STA #0041
007D 20FAFD POLLKB ; WAIT FOR MISSILE FIRE
0080 205000 JSR #0050 ; INITIATE SOUND
0083 A540 LDA #0040
0085 18 CLC ; DECREASE FREQUENCY
0086 6901 ADC #0001
0088 8540 STA #0040
008A 900E BCC MOVE ; DECREASE NOISE FREQUENCY
008C A541 LDA #0041
008E 18 CLC
008F 6901 ADC #0001
0091 8541 STA #0041
0093 C90A CMP #0010 ; MINIMUM FREQUENCY REACHED
0095 D003 BNE MOVE
0097 4CAA00 JMP #00AA
009A MOVEA540 LDA #0041 ; ALTER TIMER ONE'S LATCHES
009C 8DC6BF STA #BFC6 TO ALTER FREQUENCY OF
009F A541 LDA #0041 SOUND PRODUCED ON
00A1 8DC7BF STA #BFC7 RECORDER SOCKET OUTPUT,
00A4 206C00 JSR #006C ; DELAY
00A7 4C8300 JMP #00B3 ; REPEAT DECREASE IN FREQUENCY
00AA A700 LDA #0000 ; STOP SOUND MISSILE
00AC 8DCBBF STA #BFCB FIRED AND FLOWN AWAY
00AF 8DC2BF STA #BFC2
4CZ700 JMP #0077

```

With slightly more complicated programming, varying the frequency and sound time, overhead flying missile sounds can be made, laser sounds, sirens and even fast travelling cars!

As you can hear, once you have started the square wave output on PB 7, to vary the frequency just load the latches of Timer 1 with different values. By varying the frequency and the time, certain notes are generated. Some quite wonderful noises can be produced by an inexpensive method. All you need is a computer!

David Cawthorne 40 Westbourne Road, Westkirby, Wirral, Merseyside L48 4DH.

❖ ❖ ❖ ❖ **ORIGINAL PROGRAMS WANTED** ❖ ❖ ❖ ❖

Business – Educational – Games

We will pay up to £100.00 – Sub-routines also bought.



Second Hand Equipment Bought, Sold & Exchanged

ZACRIDE LTD.

16 IDDESLEIGH ROAD, CHARMINSTER,
BOURNEMOUTH, DORSET. BH3 7JR. 0202 294393

Sponsors of TANGERINE USERS GROUP LTD.

T.U.G. is about to launch its own HIGH DEFINITION GRAPHIC BOARD!
BLACK & WHITE..... 256x256

We are now under production for the pcb. We shall be producing the complete assembled product, or the basic pcb for the constructors. As yet we have no prices available due to the early release date of this information, however we do expect that our prices will be very favourable to the members and certainly below that of any other of its kind found on the market.....

SO GIVE US A TANhardring and keep in touch.....
OTHER PRODUCTS ON THE WAY....

FROM THE SAME STABLE THAT GAVE YOU 'ASTEROIDS' comes 'SHUTTLE LANDER'. A FULL GRAPHIC game in machine code for ^{7K} of RAM. Take control of the Shuttle in this simulator and guide your craft to a safe landing in adverse conditions of high or low wind speeds. Full Rudder and Elevator control, Left, Right & Normal Rudder control with an Artificial Horizon Indicator and Direction Indicator. Add that to three Elevator Level controls and stir briskly with three UP Elevator controls and you've got yourself feeling like your in the real thing. NINE guidance controls in all..... KEYBOARD CONTROL.....

NOW ADD THAT.....TO 3D NOUGHTS & CROSSES (from the same stable) and you've got yourself an evenings entertainment for the whole family.

CASSETTE only..... £10.00 including p.&p. & VAT.

YOU'VE USED...: SPACE INVASION & ASTEROIDS AND YOU THINK THEIR GOOD.. THEN TRY....

DIVE BOMBER

KEYPAD CONTROL

A^{7K} machine code prog with FULL graphics (this one beats me every time!) Bomb you enemy's missile defences with three advanced type aircraft with computer controlled missiles to help you. Fly your aircraft fast or slow, high or low to evade oncoming enemy missiles. But watch it! You've only got limited fuel and missile supplies unless you're prepared to dock with the Mother ship or bomb your self a landing strip. (I never get that far! I'm always crashing or getting wiped out) You don't have to be a Jump Jet pilot to enjoy this game, it does help though.....

CASSTTE only..... £10.00 including p.&p. & That VAT.....

JUST A MINUTE, I HAVEN'T FINNISHED YET... Please notice, I don't make so many (HUH!) typing mistakes as I used to, thats 'cause I been practicing with the new

TANTYPE Touch Typing Tutor.

BASIC

This comprehensive package contains eight routines including cassette stored data handling routines for full touch typing control for the Microtan keyboard. So come on, You two fingered keyboard jockey's, this is the modern way to learn.

CASSETTE only..... £8.00 including p.&p. & VAT.....

YOU'D BETTER HAVE YOUR OXFORD ENGLISH DICTIONARY READY FOR THIS ONE.....

M	This game is for the whole family to join in with.
I	
SCRABBLE	Two word idiots are required to play each other with the computer doing the sorting just to confuse things.
R	
O	I must admit, I do like this one.....
T	TUGON!
A	F
N	F
	!

CASSETTE only..... £6.00 including p.&p. & that VAT.....

STOP IT AGAIN! 24K EPROM BOARD AVAILABLE VERY SHORTLY

***** GIVE US A TUGRING AND STAY ON TOP *****

ADDITIONS

In addition to the overleaf, the following information has been added:-

DISCOUNTS

10% WBM Business Supplies Ltd Tel: Byfleet 52245 SUE AYRES

Offer a range of Printers and supplies for general and word processing applications.

Midwich Computer Company Ltd. Tel: (0284) 701321

Offer a wide range of computer and electronic components for computer Groups at reduced prices equivalent to bulk purchase prices. Send large A4 S.A.E. for complete lists
Midwich Computer Company Ltd. Hewitt House, Northgate Street. Bury St Edmunds
Suffolk IP33 IHQ

MEMORY MEMORY MEMORY MEMORY MEMORY MEMORY MEMORY MEMORY MEMORY MEMORY MEMORY
2114's 4116's 2716's etc etc etc.....

TUGRAM! (what a name!) Anyway, the constant price fluctuations on the RAM market make it very difficult for us to quote prices, as by the time you actually read this, the prices may have changed for better or worse. If you would like to give us a ring, We will quote accordingly. Our prices will remain very competitive with other distributors. SO GIVE US A TANRAMring sometime..... 0202 294393.....Bob.....

SOFTWARE LIBRARY

We just can't keep up with the constant change in our library listings.

SO GIVE US A TUGSOFTing sometime.....

EPROM PROGRAMMER

T.U.G. is now launching!! its own eprom programmer. The basic format for the programmer is as follows;

A simple programmer for the 2716 and its equivalents. The programmer has been designed for simplicity and cheapness, as an alternative to those already found on the market. It has been designed for the Microtan System on the Microtan System from the start.... We are producing the P.C.B. (Single Sided), The Construction Notes, Instructions for construction and Programming tips along with a powerfull Software Package for automatic programming from direct memory contents. The Programmer is powered by three 9 volt PP3 batteries (27v) and is connected to the three parallel ports on Tanex A1 - B1 - C1, using TWO 6522's Via's.....

Six software commands will allow you complete control over programming.....
BLANK EPROM ? / PROGRAMME EPROM / LIST CONTENTS / VERIFICATION OF CONTENTS / TEST CONTENTS / FULL SCREEN MONITORING DURING PROGRAMMING / ABORT.....

We will not be supplying the components for assembly as there are only half a dozen required for the pcb, some ribbon cable and plugs. Most of the required components will be found in the constructors spares box, besides, those which may be required, can be obtained from any of the distributors offering TUG members discounts.....

The EPROM PROGRAMMING PACKAGE we feel can be constructed by anyone that has a modicum of soldering experience.....

EPROM PROGRAMMER PACKAGE (less components) £17.00 including VAT, p&p.

Please note that the programmer requires TWO 6522 VIA's

MORE AND MORE HARDWARE PRODUCTS ARE COMING ON STREAM.. SO KEEP IN TOUCH.