Einstein Magazine

& ALL MICRO NEWS Number 101

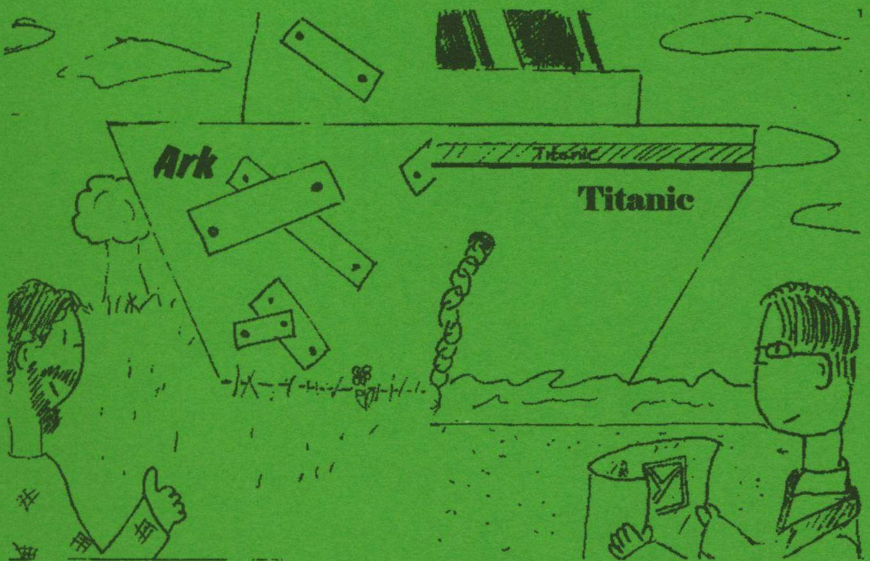
Published for users of Einstein (and other) computers

by RPM Society.

Publisher and Secretary:-

A E Adams, Ivy Cottage, Church road, New Romney,

KENT TN28 8TY



Built by Amateurs <|> **Built by Professionals**

Hastings
COACHES

COACH HOLIDAYS with

May – October 1988

DEVON COAST COUNTRY CLUB
8 days — £115
Saturday 7 May 1988

PARIS WEEKEND
3 days — £62
Friday 20 May 1988

FORT WILLIAM
7 days — £179
Sunday 29 May 1988

THE WEST COUNTRY
7 days — £169
Sunday 5 June 1988

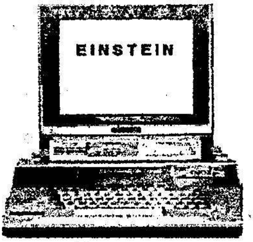
EDINBURGH
7 days — £189
Sunday 12 June 1988

FORT WILLIAM
7 days — £179
Sunday 26 June 1988

THE WEST COUNTRY
7 days — £169
Sunday 3 July 1988

NORTH WALES
5 days — £119
Monday 4 July 1988

EDINBURGH
7 days — £189
Sunday 10 July 1988



Einstein Magazine

& ALL MICRO NEWS

Number 101

Published for users of Einstein (and other) computers
by RPM Society.

Publisher and Secretary:
A E Adams, Ivy Cottage, Church road, New Romney,
Kent TN28 8TY
****

Editor: Bob Deeley
33 Britain St., Dunstable, Beds. LU5 4JA
****

Software Library, Shows & Useful Bits:
Steve Potts 85 Thorold Ave, Cranwell Village, Lincs.
NG34 8DS s.p.potts@talk21.co.uk
****

Einstien Webb Site:
http://members.tripod.co.uk/~tatung_einstien
ftp://tatung_einstein:gwcdrw@ftp.tripod.co.uk/
Ken Ross can be contacted by his email...
cbm8032@bigfoot.com

Contents

8-bit Memories

John Marriott - April 2001

Looking back I've come to realise that the moment an ATARI STfm 512K came into my life - I stopped programming! Okay, I used some programs but only in the way of functionality...

...perhaps unwittingly I'd discovered my level of incompetence' without realising what I'd discovered - but it was 'faster, bigger, brighter, better' than my SPECTRUM+3, yet somehow those 8-bitters seemed to have more comfort, more pull than the 16-bitters as I discovered ailing Spectrum Fanzines to write programs for, wallowing in the feeling of 'a one eyed person in the Kingdom of the Blind is King' yet succumbing to the Siren Call of the 'Big Blue' and the power of the 286...

..,which I've never ever done any programming on, being echoed through the 386's and on up to the Pentium. Like a lot of people I acquired various emulators - full of intention, never fulfilled - as I began to realise that part of the old 8-bitters was the 'feel', the learnt' key presses - even the touch of the keyboards. My current AST PC has a very nice one, but feels like 'dead fish' to me, so I've an old IBM clattering away on it - much the pity that I can't fix it up on my TC01 - which brings me right back to that!

Part of its appeal is its lack of memory - sounds stupid, but I 'know* that I can sit down and start hacking some other Programmer's work, break off and continue later without losing the thread of what I've been doing. I can learn' because subconsciously I recognise that I'm only 'reading' a short story, rather than Tolstoy's 'War & Peace' where little by little the 'what's the point' can creep in as you lose the plot, the thread of events...

...and it's a Programmer's Dream Machine - what with its 'multi-tasking', yes, its multitasking! Tell me, what other computer can slip so easily and almost seamlessly between MOS, DOS and BASIC, almost cry out for you to go hacking into programs, nose amongst disk files - look at other's coding, not even the BBC-B with all its pseudo-educational support, add-on ROM/RAM boards, bank-switched memory scrabbling to get over in-built ROM system limitations appeals as much.

No, this isn't an attempt to start a 'us and them' war, for I've 2 BBC Compact computers with lots of ROM images and what have you, but the 6502 chip has never appealed to me as much as the Z80 - which seems strange (looking back) for despite a 'slight' dalliance with the ZX80 and its dreaded 'RAM wobble' my next computer was a VIC-20 with masses of cartridges of RAM/Utilities/Games, and once I'd laid my hands on its Programmers Reference Guide really went to town programming,

constructing light pens and what have you, yet never got to grips with m.code because I found the 6502 mnemonics 'awkward'...

...horses for courses, for others have found the reversel Which of course brings me to the crux of the matter - in truth, what can the latest Pentium 3's do which your TC01 can't? Can you out-type your TC01, can you outmaths it, and when you're really at the palm drenching, breath stopping, teeth grinding stage of Galaxians with the music coming to a crescendo, really take in how 'blocky' and 'jerky' the screen display is - and the answer is a resounding "...NO}..."

To put it simply, your ancient 8-bit computer is waiting for you at least 90% of the time - even 99% of the time, your latest 'all singing, all dancing, take 5 minutes to initialise on start up' PC spends more time actually doing nothing as far as 'interfacing' with the User, most times 99.999% doing nothing.,

...almost like the old 'pencil & pad' by the telephone - now that's progressl

Don't junk that old 8-bitter, it's therapy, it's relaxation, it's part of our history, which is still concurrent. My eldest son has a top of the range Apple Mac at work for his 'creative design' position, the latest Sony Play station at home yet it doesn't stop him sneaking into my 'computer room' (his old bedroom) to play on an Atari 2600 games consul - and graphics don't come any 'blockier' than that! My younger son is a Computer Programmer at a local Insurance Company Head Office - rewriting COBOL programs to SQL (even Noah knew how to program Common Business Orientated Language - although he preferred Binary!) - yet relaxes with an acoustic guitar, amongst BMX stunt riding, weight training, womanising, eating & etc.

Funny, but what I've just said is - today's computers are tools, tools we use at work, tools we don't want to take home with us at the end of a working day. Our TCO1's, 8-bitters are our hobbies, our relaxation, our 'blanket comforters'. Perhaps they are also tools, but they're 'our' tools - something we want to use because we 'want' to, not 'have' to...

...perhaps that accounts for why so many EM readers are content to take out a magazine subscription, whilst others are only content to do it with articles, letters, feedback - we all have our individual needs?

—@@@—

STAFFORD SHOW, NOVEMBER 2000

By Stephen Potts

"The weather was against us!" came the cry.

How right they were. With floods all over the country and the train services not running, I was surprised to get a call asking if I was still coming to the show, considering Stafford was underwater according to the radio. Well, Graham Betany called and said all was still on and there had been a breakdown in communications at their end, so I did not get the 750 flyers to distribute as per normal. I do this as a gesture of goodwill on our "Einstein" behalf.

Well, with the invite coming late on and my heavy work commitments, etc, the year just seemed to fly by and the show sneaked up on us.

So I took a day off work on the Friday and loaded my bits and bobs into the car. I now have a diesel estate so that cost of running is less in fuel to the shows, etc. I have a big trip to do soon, so I can absorb this as my duty to the Einsteins.

On the Saturday I got up at 5.00 am to meet John my friend to assist on the stand. We drove from Lincolnshire through Newark-on-Trent. The river had burst its banks and the flood plains were full, but the road was okay apart from the odd pool across to splash through.

On arrival, we were greeted by Graham Betany and allocated a couple of tables. We unloaded quickly and took the car back out of the hall. On returning, we set three TC01's up on a range of drives, internal and external, some with 31/2" drives, others with 80 column cards.

We had Apollo on one set-up and the standard welcome disk on another. On the third I ran my favourite program, Infofax. With this program I can make a full colour slide show with animation to give a good running demo.

Infofax is part of the Infobase suite of programs. The first surprise at this really good software is the changing of the standard character sets to another set at the stroke of a single key.

This done, we can use the varied sets to produce coloured pages with background and large letters as well as graphics. I have created a few pages to try and once the style is mastered I find it very rewarding to sit back and view the finished pages as they scroll by. Perhaps a review is called for.

Well, back to the show and the floods. A few traders did not show, but the halls were full of all sorts of technological delights and we had a good spot near where the bring and buy used to be held, but I'm afraid legislation has made this a no-go item for future shows.

The doors opened and a stream of people headed in to spot the bargains and what bargains there were, for the weather had put the brakes on the attendance with only half the expected turn out. Some items were going very, very cheap. I bought two 31/2" external drives in boxes at £2.50 each and memory for my PC was half the usual price. I also bought some odds and sods, including a CD player, so that in the end I was well overdrawn on my self-imposed limit.

—@@@—

After Drive Call

(Re-write following phone conversation with Bob and a strange anomaly in one of the tables, which I had noticed but not followed through!).

Booting up in DOS 1.31 which had a drive configuration byte of &AC (0: 40T/SS 1: 80T/SS 2: 40T/DS 3: 80T/DS) then back Into MOS followed by TF9C8F9FF<e> and TFD80FDBF<e> then a screen dump, provided table 1, and is a simplistic way of showing how DOS 1.31 handles 'drive specifications' (a form of 'indirect jumping' - going to a place in RAM to collect a value to 'go off to' or 'use as a store') - the values in &FD80 to F08F being the result of DIR 0: <e> with table 2 the result of DIR 1: and table 3 of DIR 2: - but I've not been able to correlate the information gained at this time?

Whilst this section is now written with hindsight, gut feeling was telling me that later versions of DOS would have more to offer, so I tried a similar test with DOS 2.05 with table 4 showing me that there were different RAM locations twixt table 1 (which now had the suggestion that locations

&FD80/90/AO/BO in DOS 1.31 could be pointing to 'file header information' taken off the disk on a DIR), so what else was I going to find? Well, table 5 quickly hove into view - being the 'drive specification information' (listed so as to show direct comparison of drives) of DOS 2.05 and of course I instantly and wrongly jumped to the conclusion (as &28=dec40, &50==dec80 and coupled with lingering memories of my initial DOS 1.31's disk's drive configuration) for in fact table 5 refers to 40T/SS, 40T/DS, 80T/SS and 80T/DS. Strangely enough, I've not been able to find something similar in DOS 1.31 - but must admit that it's been a 'shallow scan' although any pattern as provided by table 5 should be fairly easy to spot?

Anyway.. .clear disks, switch off, restart in DOS 2.05 (drive configuration &EE) and read* in the Drive 1: 80T/DS to change that configuration byte to &FF (part of my test was to show ALL drives as 80T/DS) hopefully disassemble the *drive program* so that &FF lands up in the 'right place at the right time* (not RAM Addresses &F9CA/DA/EA/FA as I initially thought!) - will it work...

```
Table 1>TF9C8F9FF
     F9C8 00 FD EE FA 80 FO CO FD }nz}@}
     F900 00 00 00 00 00 00 00 00
     F908 00 FO EE FA 90 FD CC FD }nz.}L}
     F9EO 00 00 00 00 00 00 00 00
     F9E8 00 FD EE FA AO FD D8 FD }nz }X}
     F9FO 00 00 00 00 00 00 00 00
     F9F8 00 FD EE FA BO FD E4 FD .}nz0}d}
     >
Table 2>TFD80FDBF
     FD80 06 48 D1 41 9C 2D OE 80 HQA,-
     FD88 80 80 80 80 80 80 80 80 ..
     FD90 00 00 00 00 00 00 00 00 ..
     FD98 00 00 00 00 00 00 00 00 ..
     FDA0 00 00 00 00 00 00 00 00 ...
     FDA8 00 00 00 00 00 00 00 00 .........
     FDB0 00 00 00 00 00 00 00 00 .........
     FD88 00 00 00 00 00 00 00 00 .........
     >
```

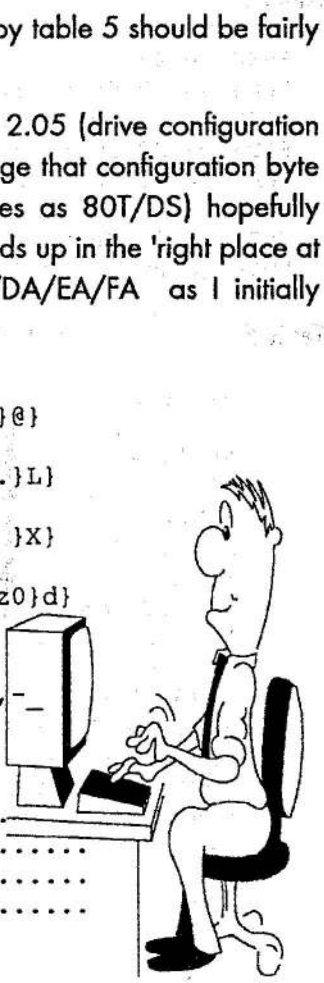


Table 3

```
>TFD80FDBF
FD80 06 48 01 41 9C 2D OE 80 .HQA.-..
F088 80 80 80 80 80 80 80 80 ........
F090 29 A3 28 2C AB CB EA 92 )£(,+Kd.
FD98 0F 3B 1C 88 80 80 80 80 .;......
FDA0 00 00 00 00 00 00 00 00 ........
FDA8 00 00 00 00 00 00 00 00 ........
FOB0 00 00 00 00 00 00 00 00 ........
FOB8 00 00 00 00 00 00 00 00 ........
>
```

Table 4

```
>TFD80FOBF
F080 06 48 D1 41 9C 2D OE 80 .HQA.-..
F088 80 80 80 80 80 80 80 80 ........
FD90 29 A3 28 2C AB C8 E4 92 3E(,-tKd
F098 OF SB 1C 88 80 80 80 80 .;......
FDA0 DE 41 DB 92 BE 8A 51 ID "A[,>.Q.
FDA8 57 AA 70 E8 01 7A 1A EO W~phOz.m
FDB0 00 00 00 00 00 00 00 00 ........
FDB8 00 00 00 00 00 00 00 00 ........
```

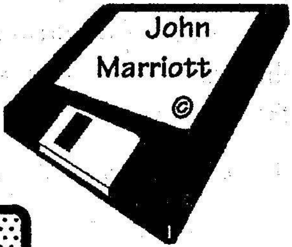


Table 5

```
               ***  EINSTEIN ***

XtalDOS 2.05        (C) Xtal 1983-87

No File
DO?
0:MOS
MOS 1.21
Ready
>TF969F9A8
F969 00 00 11 00 00 00 00 00 ........
F971 00 FD A9 F9 80 FD CO FD .})y.}@}
F979 00 00 00 00 00 00 00 00 ........
F481 00 FD D6 F9 90 FD 90 F7 .}Vyl}.w
F989 00 00 00 00 00 00 00 00 ........
F991 00 FD 06 F9 AO FD C2 F7 .}Vy }Bw
F999 00 00 00 00 00 00 00 00 ........
F9A1 00 FD 06 F9 BO FD F4 F7 .}Vy0}tw
>
```

—@@@—

THE EXPANDED GRAPHIC

By Dave Arts

Alberts Screens
Continuation from
EM 100

In order to produce an expanded graphic image, we must first join together the basic graphic image definition produced by 8x8 pixels. FIG 4 shows what I mean.

The first square starts at address &00F0,the second at &00F8, the third at &01F0 and the fourth at &01F8. The data for this Christmas tree obviously occupies more memory than for a single graphic figure, and the data should again be entered at &5000 as before.

First: The Pattern Data, in order.

```
&5000   00 01 01 03 03 07 07 0F    (GRAPHIC 1)
 5008   00 80 80 C0 C0 E0 E0 F0    (GRAPHIC 2)
 5010   0F 1F 1F 3F 7F 03 03 03    (GRAPHIC 3)
 5018   F0 F8 F8 FC FE C0 C0 C0    (GRAPHIC 4)
```

Correspondingly the attribute data follows on and if it is Green on Yellow like before we will have:

```
&5020   CB CB CB CB CB CB CB CB    (GRAPHIC 1)
 5028   CB CB CB CB CB CB CB CB    (GRAPHIC 2)
 5030   CB CB CB CB CB CB CB CB    (GRAPHIC 3)
 5038   CB CB CB CB CB CB CB CB    (GRAPHIC 4)
```

And the screen addresses follow next:

```
 &5040  00 50 F0 00 08 50 F8 00   Pattern Data address
and
  5048  10 50 F0 01 18 50 F8 01   Pattern Screen address

  5050  20 50 F0 20 28 50 F8 20   Attribute Data address
and
  5058  30 50 F0 21 38 50 F8 21   Attribute Screen address
```

The only real change to the machine code software we have to do now is instruct it to call the SETADDR sub-routine 8 times instead of 2 times for the single graphic. (4 times for the pattern and 4 times for the attributes). Therefore from basic POKE &8101,&08 or pop into MOS and change &8101 to 08 with the M command, they both do the same and that's all we need to do!

Run the Basic programme again that we entered for the single graphic and a 2x2 graphic image will appear at the top RIGHT HAND corner of the screen.

Things to do:

a) Change the attribute data for the trunk by changing the content of addresses &5035-&5037 from CB to 6B and similarly &503D-&503F from CB to 6B.

b) Change the position of the tree by altering the Data in the table starting at address &5040 viz:

```
&5040 00 50 60 06 08 50 68 06
 5048 10 50 60 07 18 50 68 07
 5050 20 50 60 26 28 50 68 26
 5058 30 50 60 27 38 50 68 27
```

NOTE only the screen addresses need to be changed, the data addresses are ok since we are not changing the addresses of pattern data starting at address &5000 or attribute data starting at address &5020.

c) Create your own 2x2 Graphic and attributes.

For those who want to create a 2X3 or 3x2 graphic image see FIG 5 and FIG 6,then the procedure is as follows:

Define the patterns; enter the data for the patterns in order starting at address &5000 (&5000-&502F)

Enter the attribute data immediately after the pattern data (&5030-&505F)

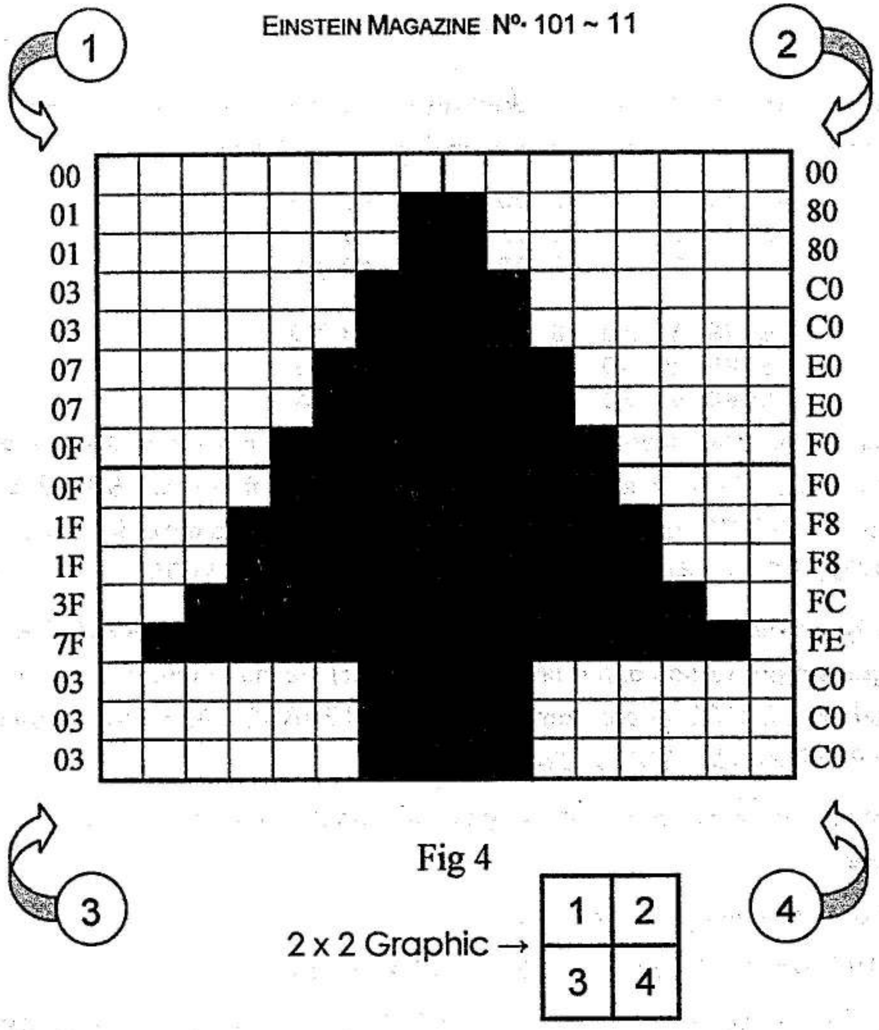


Fig 4

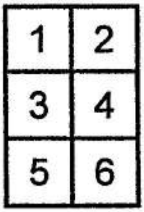


Fig 5

2 x 3 Graphic

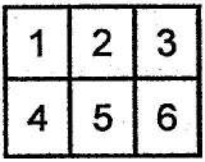


Fig 6

3 x 2 Graphic

Enter the required screen and data addresses below this; an example is shown below with addresses corresponding to a 2x3 graphic.

```
&5060 00 50 F0 00 08 50 F8 00
 5068 10 50 F0 01 18 50 F8 01
 5070 20 50 F0 02 28 50 F8 02

 5078 30 50 F0 20 38 50 F8 20
 5080 40 50 F0 21 48 50 F8 21
 5088 50 50 F0 22 58 50 F8 22
```

Because we have moved the start address of this last table from &5040 to &5060, we will have to modify the control routine at address &8103 to place the LS BYTE as &60 and not &40. Do this as before either from basic by POKE &8103, &60 or by using the M command from MOS.

Finally, as we now have to call the SETADDR sub-routine 12 times (6 times for pattern production and 6 times for attributes) we must adjust the counter at address &8101 in our control routine to 12 (&0C). Again from basic with POKE &8101, &0C or directly into MOS

Running the same Basic programme will again generate the Graphic Image.

The Data can be saved, by typing:

```
SAVE "DATASET.OBJ", &5000,  &50FF <enter>
```

and the SETADDR subroutine, PRINTOUT subroutine and CONTROLR routine can be saved together by typing:

```
SAVE "SCREENPT.OBJ", &8000,  &81FF <enter>.
```

They can be loaded back by typing:

```
CLEAR &5000:LOAD "DATASET.OBJ"
CLEAR &8000:LOAD "SCREENPT.OBJ"
```

The small basic extension can then be entered and run.   Happy tapping!!

—@@@—

3DBAR.BAS

QBASIC PROGRAM

By Dave Williams©

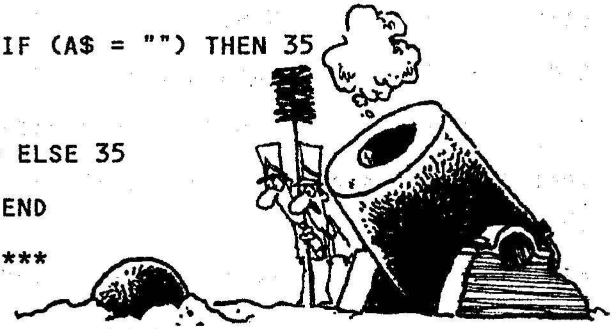
This is a 12 cycle auto-ranging barchart whose bars are displayed in a 3D format to enhance their appearance. Errors in the input can be deleted quickly at the time they are made although I have been informed that members do not make mistakes?

```
REM *****************************************************
REM *             AUTO-RANGING 3D BAR CHART             *
REM *      Written for PC (QBAS) by D. Williams         *
REM *                updated Jan 2001                   *
REM *****************************************************
5 CLS : CLEAR : SCREEN 9: COLOR 15, 0: DIM Y(13): LOCATE
3, 28
PRINT "AUTO-RANGING 3D BAR CHART": DRAW "C15BM216,42R200"
A$(1) = "100%": A$(2) = " 75%": A$(3) = " 50%": A$(4) = "
25%"
A$(5) = "  0%": DRAW "BM7,5R625D339L625U339"
REM ******** DRAW WINDOW, INCREMENTS & LABEL AXES ********
DRAW "BM511,57D8NR5D114l1NU122L388U2NL5U28NL5U28NL5U28NL
5U28NL5U8R1ND122"
FOR F = 1 TO 5: LOCATE F * 2 + 3, 11: PRINT ; A$(F): NEXT
FOR F = 0 TO 352 STEP 32: LINE (138 + F, 180)-(138 + F,
183), 15: NEXT
FOR F = 1 TO 12: LOCATE 14, 13 + 4 * F: PRINT ; F: NEXT
FOR F = 1 TO 5: LINE (127, B + 65)-(507, B + 65), 14, ,
&HF0F0
B = B + 28: NEXT: B = 0
```

```
REM ****************** INPUT VALUES ********************
LOCATE 23, 19: PRINT "KEYS: Input numbers / D = Delete any
errors"
DRAW "BM144,324C15R345"
FOR F = 1 TO 12: A = A + 1: IF (A = 7) THEN A = 1: B = 24
B$ = LTRIM$(STR$(F)): I$ = "0" + B$: IF (F > 9) THEN I$ =
B$
10 LOCATE 15 + A, 35 + B: PRINT SPC(45 - B); : L = 0: t =
0: C$ = ""
15 LOCATE 15 + A, 35 + B + L: PRINT "?": FOR W = 1 TO 800
IF (W > 400) THEN LOCATE 15 + A, 35 + B + L: PRINT " "
NEXT W: LOCATE 15 + A, 20 + B:
PRINT "Input No: "; I$; " = ";
A$ = UCASE$(INKEY$): IF (A$ = "") THEN 15 ELSE D = ASC(A$)
IF (D = 8 OR D = 68) THEN 10
IF (D = 27) THEN CLS : COLOR 7, 1: END
IF (D = 13 AND C$ <> "") THEN 20
IF (D = 47 OR D < 46 OR D > 57) THEN 15
IF (D = 46) THEN t = t + 1: IF (t > 1) THEN 15
C$ = C$ + A$: Y(F) = VAL(C$)
L = LEN(C$): IF (L > 6) THEN 10
COLOR 14: LOCATE 15 + A, 34 + B + L: PRINT A$: COLOR 15:
GOTO 15
20 C = C + Y(F): IF (Y(F) > N) THEN N = Y(F): REM (N=max
input value).
NEXT F: B = 0: C = C / 12: IF (C <= 0) THEN 25 ELSE Z =
(112 / N) * C
```

```
COLOR 12: F = 12 * C: LOCATE 14, 66: PRINT "Total input":
LOCATE 15, 66
PRINT "="; F: C = INT(100 * C + .005) / 100: COLOR 14
LOCATE 4, 66: PRINT "Peak input": LOCATE 5, 66: PRINT "=";
N
COLOR 11: LOCATE (184 - Z) / 14, 66: PRINT "Average input"
LOCATE (198 - Z) / 14, 66: PRINT "="; C
LINE (124,177 - Z)-(509,177 - Z), 11: D = 16: E = 8: A = 0
REM ************** DRAW COLUMNS ****     ←  same as REM
FOR F = 1 TO 12: Y(F) = 112 / N * Y(F):' Y Amplitude (Auto corrected).
PSET (145 + 2 * D * (F - 1), 177): Y = Y (F): IF Y= 0 THEN
25
A$ = "C15L=" + VARPTR$(D) + "U=" + VARPTR$(Y) + "R=" +
VARPTR$(D)
A$ = A$ + "D=" + VARPTR$(Y) + "E=" + VARPTR$(E) + "U=" +
VARPTR$(Y)
A$ = A$ + "L=" + VARPTR$(D) + "G=" + VARPTR$(E) + "R=" +
VARPTR$(D)
A$ = A$ + "E=" + VARPTR$(E): DRAW A$
25 NEXT: A = 0
REM ***************** PAINT COLUMNS *******************
FOR F = 140 TO 492 STEP 32: A = A + 1: IF (Y(A) = 0) THEN
30
PAINT (F + 6, 175), 1, 15: REM Side colour.
PAINT (F, 176), 2, 15: REM Front colour.
PAINT (F + 6, 175 - Y(A)), 9, 15: REM Top colour.
30 NEXT
```

```
REM ****************** DECISION TIME ******************
FOR F = 1 TO 2: LOCATE 22 + F, 18: PRINT SPC(50); : NEXT F
COLOR 15: LOCATE 23, 28: PRINT "KEYS: R=Repeat or E=End"
DRAW "C15BM216,324R184"
35 LOCATE 23, 52: PRINT "?": FOR W = 1 TO 1000
IF (W > 500) THEN LOCATE 23, 52: PRINT " "
NEXT W
A$ = UCASE$(INKEY$): IF (A$ = "") THEN 35
IF (A$ = "R") THEN 5
IF (A$ = "E") THEN 40 ELSE 35
40 CLS : COLOR 7, 1: END
REM ******* END ********
```



—@@@—

Musings on the AABBBBAA Switch

Back in EM94/9-13 Ted Cawkwell published an article on 'electronicifising' Chris Coxall's mechanical switch (EM89/17-20) - and whilst personally I felt it a 'bit over the top', for despite my 'electronic credentials' I'm much of the opinion that 'simple tin bashing' is best - but his article started me thinking, mainly on EM94/12 'the second way' section when I thought 'why can't you have the disk on boot-up load the Function Key(s) so that by <FO> a 'OUT &32,0<e>:DIR 0:<e>' and <F1> 'OUT &32,1<e>:DIR 0:<e>' so allowing a double sided boot drive to be software hardware side switched, but the action being transparent to the User...

...such simple ideas - but what a learning curve, and still learning!

With a copy of your original DOS 1.31 disk start your TC01 without any disks in and at the MOS prompt (with disk now in) do R01001000<e> then T0100016E<e> and you should get a screen display as (fig. 1) if it is original and 'unmodified'. Type M0162<e> and the value should show as C9. Enter the values from (fig.2) 0162 and at 01 A8 the <.><e> and T010001AE<e> and check/compare it against (fig.2) listing - and if correct save it back with W01001000<e>.

Now do a <ctrl><break> to reboot and at the 0: prompt press <FO>. The listings (fig.3) and (fig.4) were the result of my pressing <FO> and <F1>. Whilst they showed my program functioned, it took me a little while to comprehend why my 'call' command wasn't working as I expected - seeing as I was in DOS with an XBAS command? Try it and see, part of your learning curve?

What those extra bytes have done is to poke values into VRAM and a couple of m.code 'returns' in &8000 and &8010 RAM (as - I want you back here for the next command') from the <FO>/<F1> keys, the simplistic idea being that at &8000 would be my OUT &32.0 and at &8010 my OUT &32.1 m.code commands as envisaged in Ted's program. The disassembly listing is shown to clarify a certain point;

```
call  0162  3E C9       LD A, C9       :data - RTN from 'function'
      0164  21 00 80    LD HL, 8000    :RAM location
      0167  77          LD(HL), A      :and put it into RAM
call  0168  3E CS       LD A, C9
      016A  21 10 80    LD HL, 8010
      016D  77          LD(HL), A      :ditto
Fkey  016E  3E 0        LD A, 80       :VRAM low byte
      0170  D3 09       OUT(09), A     :& send it
      0172  3E 7B       LD A, 7B       :VRAM high byte
      0174  D3 09       OUT(09), A     :& send it
      0176  21 82 01    LD HL, 0i 82   :data start in RAM
next  0179  7E          LD A, (HL)     :get data into 'A'
      017A  D3 08       OUT(08), A     :'stream's set up, send!
      017C  23          INC HL         :next data position in RAM
      017D  FE 00        CP 0          :is it the 'end of data zero?'
      017F  20 F8        JR NZ next    :no, so loop bac~ & action
      0181  C9           RTN           :yes, so go back from where
                                       :you came from!
data  0182                             :required datal
```

Point to make - some of you will spot that the Function Key VRAM starts at &3B80, so why the &7B80 figure? Well, The Source' states that &40 must be added to the 'high byte' figure to ensure the VDP recognises that a write to VRAM request is on its way - that little point took me two days of scanning just about any/or everything I had on the TC01 as I kept on 'wondering' where my data was going to when I steadfastly kept to &3B80l

The loop' putting the data into VRAM isn't very elegant, but it works. In the same vein in 'caU1' and 'caU2' there is no need to 'LD A, C9* twice with the above coding, but there could be an awful lot of 'data' before that 'C9' of setting up the User Port & etc., so the value in 'A' shouldn't be taken for granted ~ what?

As the above doesn't do what I thought it should, the idea came to me of replacing the 'data' with

```
data  0182              :MENU (4D454E558D)
      0187 00     NOP   :end of data marker
end   0188 C9     RTN   :end of <ctrl><break> return
                        :(00 redundant data after)
```

the listing sets up <FO> to load a MENU. COM file from a disk (if it's on itl). Yet again this did not function as I wanted - which was to be a straight off auto-boot of that MENU.COM...

So, with the usual MOS disk read, T03000340<e> and all being well on your 'bog standard 1.31 copy disk' the display should be something similar to this

```
C3 07 E6 C3 09 El 02 80 09 00 00 00 00 etc.
```

With the M command overwrite the 00 00's with 4D454E55<„><e> and W01001000<e> it back after checking you've done that simple task correctly. The last four 'pairs' are ASCII for MENU (no need to put in .COM), so it's just as easy to put in what .COM file you want to auto-boot such as DIR, TASWORD, WP80 & etc.

The same can be done with DOS 2.02 except this appears to be at 0200 after the usual MOS disk read, except

```
C9 EC3 4 18 48 22 02 3F 03 00 00 00 etc.
```

So overwriting those 00 00's with 444952<.><e> & etc., results in an auto DIR action. Naturally - the responsibility is yours

Remember - a <ctrl><break> or the <reset button> is not the same as switching off, counting to twenty then switching on again - so if you're finding 'rubbish problems' abounding, do the latter rather than insisting on the former...

...also part of my learning curve!

Now, whilst I've not tried it - the thought crosses my mind that as in DOS the command GO takes you straight to RAM location &0100 whether you like it or not, there's a choice of paths - back into MOS, or forward into XBAS. Going back into MOS from DOS seems a strange choice, but with say <FO> set up as 'MOS<e>G8000<e>Y<e>DIR<e>' you go back into MOS, jump to your m.code routine of 'setting up the User Port and sending out that [side select nibble] and returning', back into DOS with a DIRectory call'.

At present I'm struggling to understand the setting up of the OUT &33 with the follow-on OUT &32, which causes me to heartily agree with Ted's comments of 'if it isn't broke, don't mend if. But looking back through what paperwork I've got (even Lethenthal's Z80) gives me that 'brain dead' feeling as it becomes quite obvious that 'others claiming to know - don't'. Somehow in XBAS the OUT &32.5 example doesn't seem to work for me but that might be to do with my 'expectations' of what I 'assume will happen' - yet more learning?

And that's it for the moment - quite a deviation away from what I started out, intended to resolve - but at least I now know how to program the <F keys> and create <auto program boot> disks, simply...

...yes, and good old basic 'tin bashing' beats 'complex programming' hands down?

One final point (for the moment), I 'know* that MOS disregards leading zero's, but believe me - they have a value, especially when 'writing back' to your disks as W01001000 can somehow become anything but as yet again Murphy's Law tangles your fingers, addles your brain - and in horror you realise that you've written something back to the wrong place - usually over part of an important file that you've only 'that copy of...

. . . been there, time and time again !

Fig. 1

```
>T0100016E
0100 00 E1 00 FB 00 FA 00 FF .a.{.z..
0108 04 CD 54 E1 3A 30 FA FE  MTa:0z~
0110 04 CA 03 E6 3E 01 B7 D3 .J.f>.7S
0118 23 D3 24 C3 E1 00 CD 54 £S$Ca.MT
0120 E1 CF CF OD OA 54 41 54 aOO..TAT
0128 55 4E 47 2F 58 74 61 6C UNG/Xtal
0130 44 4F 53 20 31 2E 33 31 DOS 1.31
0138 20 20 20 20 20 20 20 20
0140 28 43 29 20 31 39 38 33 (C) 1983
0148 20 31 39 38 B4 21 96 FA  1984!.Z
0150 22 68 FA E9 C5 AF 16 80 "hziE/..
0158 01 00 3B CF C3 03 15 20 ..;OC..
0160 FA C1 C9 FF FF FF FF FF zAI.....
0168 FF FF FF FF FF FF FF .......
>
```

Fig. 2

```
>T010001AE
0100 00 E1 00 FB 00 FA 00 FF .a.{.z..
0108 04 CD 54 E1 3A 30 FA FE .MTa:Oz~
0110 04 CA 03 E6 3E 01 B7 D3 .J.f>.7S
0118 23 D3 24 C3 E1 00 CD 54 £S$Ca.MT
0120 E1 CF CF OD OA 54 41 54 aOO..TAT
0128 55 4E 47 2F 58 74 61 6C UNG/Xtal
0130 44 4F 53 20 31 2E 33 31 DOS 1.31
0138 20 20 20 20 20 20 20 20
0140 28 43 29 20 31 39 38 33 (C) 1983
0148 20 31 39 38 B4 21 96 FA  1984!.z
0150 22 68 FA E9 C5 AF 16 80 "hziE/..
0158 01 00 3B CF C3 03 15 20 ..;OC..
0160 FA C1 3E C9 21 00 80 77 zA>l!..w
0168 3E C9 21 10 80 77 3E 80 >l!..w>.
0170 D3 09 3E 7B D3 09 21 82 S.>{S.t.
0178 01 7E D3 08 23 FE 00 20 .~S.~.
0180 F8 C9 43 41 4C 4C 20 26 xICALL &
0188 38 30 30 30 OD 44 49 52 8000.DIR
0190 20 30 3A 8D 43 41 4C 4C  0:.CALL
0198 20 26 38 30 32 30 OD 44  &8020.D
01AO 49 52 20 30 3A 8D 00 C9 IR 0:..I
01A8 FF FF FF FF FF FF FF .......
>
```

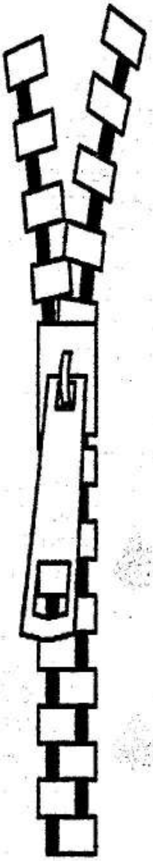


Fig. 3

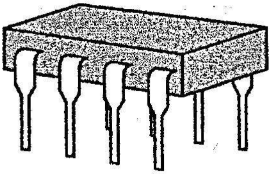
```
0:DIR O:
0:*XBAS         .COM   :*BACKUP      .COM
0:*COPY         .COM   :*INTRO       .XBS
0:*DEMO         .XBS   :*CLOCK       .XBS
0:*EINSTEIN     .XBS   :*MASTMIND    .XBS
0:*OTHELLO      .XBS   :*KEYBOARD    .XBS
0:*HANGMAN      .XBS   : SNAKE2      .XBS
0:*PICPEN       .XBS   :*DICTIONH.DAT
0:*DICTIONE     .DAT   : 0
0: AUTOEX       .COM   : COPYANY .COM
0: AUTO         .COM   : CAT .COM
0: MENU         .COM   : SLISTING.BSC
0: MUSLCFJ      .XBS
122k Size,      66k Free, 190k Total
0:
```

Fig. 4

```
      0:CALL &8020
      No File
      &8020?
      0:DIR O:
      0:*XBAS         .COM   :*BACKUP      .COM
      0:*COPY         .COM   :*INTRO       .XBS
      0:*DEMO         .XBS   :*CLOCK       .XBS
      0:*EINSTEIN.XBS        :*MASTNINU    .XBS
      0:*OTHELLO      .XBS   :*KEYBĜARD    .XBS
      0:*HANGMAN      .XBS   : SNAKE2      .XBS
      0:*PICPEN       .XBS   :*DICTIONH.DAT
      0:*DICTIONE     .DAT   : 0
      0: AUTOEX       .COM   : COPYANY .COM
      0: AUTO         .COM   : CAT .COM
      0: MENU         .COM   : SLISTING.ASC
      0: MUSICF3      .XBS
      122k Size,      66k Free, 190k Total
      0:
```

John Marriott 12th. Febuary, 2001

—@@@—



This Time Next Time

Editorial

There isn't one ... you don't want to have to hear me all the time do you. A wealth of material from John Marriott is accumulating, and his latest on the 8-bit micro says it all, so over to John ...

A little explanation of goings-on, in this issue we complete Dave Art's tutorial on Albert's Screen, defer John Marriott's *Not More Drives* series until next time, editor lost the thread – failing of most editors, John warns me! However there is an addendum/errata for first part in EM100 (See *After Drive Call*).

The Staffs All Micro Show Report was diverted via Ivy Cottage, so a bit late but no matter; it's an interesting read. A new batch of programs has been kindly sent in by Dave Williams ...remember his regular contributions? We enjoyed his fine polished application of XTAL Basic, and these are no less so but are in Quick Basic and run on the PC. Dave has been diverted these past five years, no doubt by the allure of higher definition graphics only to find Quick Basic Users have finished, but hey! We are into all sorts around here and can learn a lot, maybe adapt these programs to XTAL, and something useful on our PCs into the bargain.

Quick guidelines: Please send in contributions on a disc if you can - plus hard copy for a quick read through and formatting guideline. Any TC-01 format, MSDOS 3.5" floppy (up to 1.4Mb), CDROM: ASCII text*, Word (up to Word2000), most picture formats but avoid TIF compression. *Note. ASCII text, try not to use CR (carriage return) at line ends, use only before a new paragraph which shouldn't be indented or spaced i.e. two CRs together – saves me having to remove them all – I can spot you intention of a paragraph from the preceding single CR.

The scanner is far from perfect for OCR (optical character recognition), and is best left to diagrams and pictures. Another problem of OCR is with page breaks - I get awfully confused putting them all back together again. To facilitate mixing text and pictures: best to indicate with some standard note where you would like diagrams, marked or file named accordingly, to be inserted, e.g. <newline> *Fig 6>, *Word table 3>, *Aunt Maud photo> (See also *Post script – idea*). TC-01 screen dumps are a problem, dot matrix print outs don't scan to OCR, is there some simple way to direct them to a file? Anyone?

The Contents are not in their box, all part of Word automation process and finding out... more on what that's all about soon I hope.

Next issue we look forward to new items from Chris Coxall and continue with John's drive explorations.

Post script – idea It has just occurred to me this would work quite well when sending in small errata insertions, and showing deletions, e.g.

```
*"fox">    the quick brown *wolf^ jumped.
*Fox File> the quick brown * jumped.
```

Key: Text between *...^ is substituted,
Fox File ="fox",
_*_ Indicates a missing text only (under score = space).

All this highlights the limitations of incorporating extraneous symbols without creating ambiguity, particularly exacerbated when dealing with code. Any ideas on that one – now there's an opportunity to break the ice with a first time letter even if it's just to say what's he talking about!

—@@@—

Mr J M Taylor of Cheltenham writes that he has a twin drive Einstien with green screen and manuals, tucked away in loft, needs good home, can deliver within 20 miles. Tel:01242 602803.

PARIS WEEKEND
3 days — £62
Friday 5 August 1988

YORKSHIRE DALES
7 days — £169
Sunday 7 August 1988

WESTON-SUPER-MARE
5 days — £115
Monday 8 August 1988

FORT WILLIAM
7 days — £179
Sunday 14 August 1988

EDINBURGH TATTOO
5 days — £135
Monday 22 August 1988

AMSTERDAM WEEKEND
3 days — £72
Friday 26 August 1988

THE WEST COUNTRY
7 days — £169
Sunday 28 August 1988

NORTH WALES
5 days — £119
Monday 29 August 1988

YORKSHIRE DALES
7 days — £169
Sunday 4 September 1988

THE LAKE DISTRICT
7 days — £169
Sunday 18 September 1988

PARIS WEEKEND
3 days — £62
Friday 23 September 1988

SCOTLAND — GRAND TOUR
12 days — £289
Monday 3 October 1988

NORTH WALES
5 days — £119
Monday 24 October 1988

Local pick-ups from Hastings, Bexhill, Battle, Rye and Eastbourne.
See our colour brochure for full details of holidays listed above.
Call at any of our Travel Information Offices at:
Queens Road Coach Station, Hastings • Beaufort Road, Silverhill
M.E. World Travel, Western Road, Bexhill • Baldwins Travel, Rye
or telephone Hastings 433711 or Eastbourne 410710

ROMAN COIN HOARD FOUND IN LONDON

In January, 2001 the Museum of London announced the recent discovery of an important group of Roman gold coins found in the City. They were found during a formal excavation at Plantation Place. A hoard of 43 gold aureii coins was found buried in the floor of a large Roman house. From the residue surrounding the 'coins it has been deduced that they were placed in a leather bag inside a box before being hidden in a masonry lined hole under the floor of a substantial Roman London house, parts of which date to the later 2nd century AD. The coins were clearly concealed with the intention of recovery; that they were not reclaimed remains a mystery, particularly as the building was occupied for over a century after their deposition. This is thought to be the first hoard of Roman gold coins found in the City.

The Roman currency system was based on three metals: the aureus in gold, the denarius in silver and lower denominations in copper alloy. This hoard consists solely of gold aureii, a coin not in everyday circulation but used by bankers and rich merchants. The 43 coins are estimated to have been worth nearly four years' salary for a Roman legionary soldier.

The coins span a period of over 100 years, ranging in time from the reign of the music-loving Nero (AD54-68) to Marcus Aurelius (AD161-180). They must represent the private savings of a Roman citizen, who seems to have kept the coins as gold bullion, quite regardless of the age of the coins. Whoever buried the hoard was clearly never able to return to recover it and a whole range of interesting circumstances come to mind. It seems likely that the hoard was buried sometime in the period AD175-190.

Under the 1996 Treasure Act, the find was immediately notified to the City of London Coroner. The find has been generously donated to the Museum of London by The British Land Company plc, the owners of the site at Plantation Place in the City, who are also funding the excavation and publication of the site prior to a major new office development

This discovery is of some extra interest to Kent archaeologists and KAR readers for in 1957 (Ref l) another hoard of gold coins, this time 34 in number, was found whilst foundations were being dug at Bredgar just east of the River Medway (see also page 69 above). The coins also covered a period of about 100 years, ranging from Julius Caesar to Claudius, the latest being of AD41-42. It is generally thought that this hoard was buried by a Roman army officer just before, or after, the battle of the Medway.