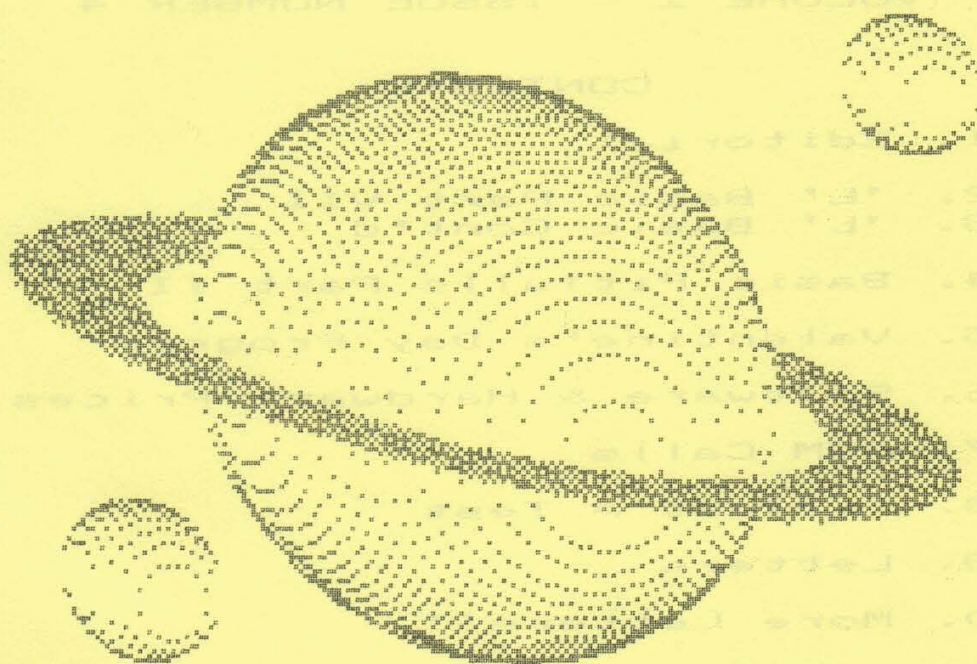


THE
MEMOTECH OWNERS CLUB
MAGAZINE



PUBLISHED BY MEMOTECH OWNERS CLUB
23 DENMEAD ROAD
HAREFIELD SOUTHAMPTON

MEMOTECH OWNERS CLUB

CIRCA ... 247

M.O.C.

VOLUME 2 ISSUE NUMBER 5

CONTENTS

- 1. Editorial
- 2. 'L' Basic Part VII
- 3. 'L' Basic Cont'd
- 4. Basic Pitfalls Part II
- 5. Valentine's Day Program
- 6. Software & Hardware Prices
- 7. RDM Calls
- 8. ... Just A Test
- 9. Letters
- 10. More Letters!!
- 11. A/D Converter
- 12. Reviews
Roller Bearing
Memosketch
- 13. Program Library

o o o o o o o o o

MEMOTECH OWNERS CLUB
 22 DENVER ROAD
 WAREHOUSES SOUTHAMPTON

E D I T O R I A L (February 1986)

Phil Eyres
23 Denmead Road
Harefield
Southampton
SO2 5BS

Richard Adams
18 Nightingale Rd
Pilands Estate
Bursledon
Southampton

Stop Press

I have just received a diskette from Christian Wohlbier (Swiss User Group) containing several interesting programs that I will include in next months program library. As a quick insight;
On the CP/M front there is a Small-C Compiler, complete with examples and documentation.

In Basic there are several very good programs:

- i) 3D-Function - Demos Saturn (This month's front cover!!)
Sinpr
- ii) Spr-Ed
- iii) Z-Wandl - Number Base Conversion Program
- iv) Memograph

The winner of this months prize draw is :-

Nicholas Hill, who will receive a copy of Dr Franky for his Rom Routines article.

...remember all you have to do to be included in this free draw is have something published in the magazine.

You may have noticed this month that there is a definite lack of assembler in the magazine, please could we have some more articles, information, programs etc. and what about something useful about CP/M since we have a growing membership in this field.

Thanks to everyone who has used our Hotline on Monday evenings between 6 & 7pm, remember we always look forward to hearing from you, the number to phone is Bursledon (042121) 5489. Ask for Rich!

If anyone would like back issues they are available for all past magazines for the small remittance of 80p. At present there are 14 back issues, 10 for volume 1 and 4 for volume 2.

It should be noted that all articles are the copyright of the sender and M.O.C., anyone wishing to have articles published elsewhere should inform us first.

---ooo000ooo---

INTERFACING PROJECTS

Why not make your new years resolution one which will lead you into the exciting world of micro electronics. Infact what better way to start than with an MOC D.I.Y. kit. Everything you need is supplied, except a soldering iron, wire cutters and of course a few hours of your time!!. So why not order now.

Interface price list

A full set of components and instructions for the LED kit. -->£6.95
A full set of components and instructions for the Speech Synthesiser kit-->18.00
Connecting cable for the internal port (needed for projects) -->£4.50
All prices are fully inclusive. Please allow 14 days for delivery and make cheques payable to MOC.

DBASE PART VII

The concluding episode in this series of 7 articles on building a Database, should leave you with a fully usable program, written in modular form, allowing you to update as ideas and techniques come to the fore. I am working on an enhancement myself, a sort, utilising the assembler sort program printed in Vol 1 issue 9.

This month, Option 4 from the main menu, the Amend Data Routine. This is very simple, involving only the 'display' of existing records, and allowing them to be changed as necessary. This infact can be done in very few lines of code, but would result in a very 'user unfriendly' routine. I therefore took the opportunity to produce a routine that is much more involved and results in a more user friendly atmosphere.

The area of program in question is the editing of the displayed string of text. I decided the concept I would use, would be to place the cursor at the beginning of the displayed string and allow it to be moved to the right, using the right arrow key on the cursor pad, having moved to the required position, change the offending characters and exit the edit, without moving the cursor to the end of the string.

To achieve this I used a few commands that up to now I have not used in the program, these are:-

INKEY#
LEFT# & RIGHT# and

The PRINT command to send non-print control characters to the screen, in this case PRINT CHR\$(30) to turn the cursor on and PRINT CHR\$(31) to turn it off.

Movement forwards and backwards through the data file FILE# is achieved using the ')' & '(' keys respectfully. For the purposes of finding the record to edit it is possible to reduce the record displayed to just the first field, this in real terms does not save much time, as displaying seems to appear instant, but it does reduce considerably, fatigue to the eyes, which has to be worth something.

Another enhancement is having the field names displayed 'right-justified', making it easier to view the records when they are displayed. Infact this particular 'bit' would make record entry clearer in the Open File routine were it implemented.

Limitations; Editing records is kept fairly simple, this was done to try and reduce the chance of program error. Pressing <RET> whilst at the beginning of a file will exit the edit, so to edit a field which is placed lower down in the record will require moving the cursor at least one character to the right before keying <RET>.

Some general notes on the programs use.

The program was originally designed as a general database, allowing a certain amount of freedom with how the structure of the database would look, as it stands, the programs maximums are:

1. Up to 6 fields, containing a max. of 30 characters each
2. Each field can have a specific name up to 30 characters long.
3. Up to 100 records maximum will fit in one file. This maximum, "should" fit in a MTX500, the program can be expanded to use the greater memory of a MTX512 but saving data files to tape would be somewhat cumbersome.

A new file is created using option 3, the Open File routine, to exit the data entry mode input an '*' in the first field of a record. It is probably best not to input too many records without saving them at intervals, better safe than sorry.

All the other routines are fairly self explanatory and should not be too much trouble to fathom out, if you play with them for a while.

It would make a good project to change the data save and load routines to suit a disc based system, as this would circum-navigate the main restriction of the program as it stands.

As ever all updates welcome, and listings of the full program are available, please send an SAE.

I will try and write a sort routine for a future article!!.

Phil Eyres


```

3997 REM-----
3998 REM AMEND DATA ROUTINE
3999 REM-----
4000 CLS
4010 IF REC(1)>1 THEN GOTO 4060
4020 CSR 12,3: PRINT "!!! FILE EMPTY !!!"
4030 CSR 11,5: PRINT "PLEASE USE OPTION 3 !"
4040 PAUSE 5000
4050 GOTO 4500
4060 CSR 7,3: PRINT " *** AMEND DATA ROUTINE ***"
4070 CSR 7,4: PRINT " ~~~~~"
4080 FOR I=1 TO FIELD(1)
4090 LET P=1: LET W$=""
4100 IF ASC(MID$(FNAME$(I),P,1))=0 THEN GOTO 4130
4110 LET W$=W$+MID$(FNAME$(I),P,1)
4120 LET P=P+1: IF P<11 THEN GOTO 4100
4130 LET OFFSET=LEN(W$): IF OFFSET>10 THEN LET OFFSET=10
4140 CSR 10-OFFSET,7+I: PRINT W$
4150 NEXT I
4160 LET RECSTORE=1
4170 LET PFIELD=1
4180 GOSUB 4600
4190 CSR 7,16: PRINT "< - BACKWARD REC"
4200 CSR 7,17: PRINT "> - FORWARD REC"
4210 CSR 7,18: PRINT "P - PART REC"
4220 CSR 7,19: PRINT "F - FULL REC"
4230 CSR 7,20: PRINT "E - EDIT REC"
4240 CSR 7,21: PRINT " (<,>,P,F,E,X...?)";
4250 PRINT CHR$(31);: CSR 28,21: LET A$=INKEY$: PRINT A$: PAUSE 100: CSR 28,21:
PRINT " ";: IF A$="" THEN GOTO 4250
4260 IF A$="<" AND RECSTORE>1 THEN LET RECSTORE=RECSTORE-1: GOSUB 4600
4270 IF A$=">" AND RECSTORE<REC(1)-1 THEN LET RECSTORE=RECSTORE+1: GOSUB 4600
4280 IF A$="P" THEN LET PFIELD=1: GOSUB 4600: GOTO 4250
4290 IF A$="F" THEN LET PFIELD=FIELD(1): GOSUB 4600: GOTO 4250
4300 IF A$="E" THEN GOSUB 4700
4310 IF ASC(A$)=88 THEN RETURN
4320 GOTO 4250
4490 PAUSE 10000
4500 RETURN
4600 REM----- PRINT RECORD -----
4610 FOR I=1 TO FIELD(1)
4620 CSR 11,7+I: PRINT " ";
4630 NEXT I
4640 FOR I=1 TO PFIELD
4650 CSR 11,7+I: PRINT FILE$(RECSTORE,I);
4660 NEXT I
4670 RETURN
4700 REM RECORD--- EDITING ROUTINE ---
4710 LET PFIELD=6: GOSUB 4600
4720 LET A=1
4730 CSR 11,7+A: GOSUB 4800
4740 IF Z=0 THEN RETURN
4750 LET FILE$(RECSTORE,A)=LEFT$(W$,LEN(W$))+RIGHT$(FILE$(RECSTORE,A),30-LEN(W$))
4760 LET A=A+1
4770 IF A>FIELD(1) THEN RETURN ELSE GOTO 4730
4800 LET I=1: LET W$=""
4810 PRINT CHR$(30);
4820 LET A$=INKEY$: IF A$="" THEN GOTO 4820
4825 LET Z=ASC(A$)
4827 IF Z=12 OR Z=8 OR Z=127 OR Z=26 OR Z=10 THEN GOTO 4820
4830 PRINT A$;
4840 PAUSE 200
4850 IF Z=13 AND I=1 THEN LET Z=0: RETURN
4860 IF Z=13 THEN RETURN
4870 IF Z=25 THEN CSR 11+I-1,7+A: LET W$(I)=SPK$: CSR 11+I,7+A: LET I=I+1: GOTO 4820
4880 LET W$=W$+A$: LET I=I+1: GOTO 4820

```

*** THE END ***

AVOIDING BASIC PITFALLS PART 2

By
Mike Pike

This is the second in a series of two articles, concerned with problems encountered when programming in Basic. This month we will concern ourselves with number manipulation from Basic.

Have you ever wondered why the Memotech won't accept that two numbers are equal when they obviously are?! Like any computer, it has to store numbers to a limited degree of precision. This is perfectly adequate for all normal home (and most business) uses but the problem can arise when you test for equality.

Try this:-

```
10 CLEAR
20 LET A=.9999999999
30 LET B=1
40 PRINT A,B,(A=B)
```

You should be surprised by the result. I was! A & B are both equal to 1 apparently. The Memotechs precision is not sufficient to discriminate to the tenth significant figure but the logical expression (A=B) gives the result zero, meaning false!!

So if this had occurred in a program, (eg IF A=B THEN GOTO 10) the condition would be false and the goto not executed even though A & B would both print out as "1". This can be infuriating when trying to test a program!

Back to the PANEL at C000 (see last month's magazine for details) and you'll see these values:-

```
C000: >C1 C2 FF 00 00 00 00 B1
C008: FF FF FF 7F 80
```

The first 3 bytes are the name table as before, this time showing 2 numeric variables. Next come the 5 byte blocks in reverse order - B first. This time these bytes do not contain pointers but actual values in a special form. How these bytes represent the numbers is not important at the moment. But you will notice at once that they are not the same, yet close enough for the print routine to come to the same conclusion about how they should be displayed.

My advice is never to use "=" in conditional expressions unless the variable was assigned a constant and is being compared with that same constant.

```
50 LET A=10
IF A=10 THEN ...
```

should always work whatever the value of the constant. But sequences like:-

```
LET A=63/9
LET B=42/6
IF A=B THEN ...
```

are 'odgy' and the conditional should be replaced by:-

```
IF A<B+.5 AND A>B-.5 THEN ...
```

You've now specified how close A & B have to be before the program should regard them as equal! It looks cumbersome but it will save a lot of headache, I promise!

The problem can arise in some unexpected places. For instance I had trouble with this statement:

```
FOR I=2 TO 1 STEP -1
```

How could that possibly go wrong? Well sometimes the loop was executed twice but occasionally only once!! Apparently when you subtract 1 from 2 you don't always get 1. By the same logic as before, the problem is solved by replacing the statement with this:-

```
FOR I=2 TO .5 STEP -1
```

This does not alter the values that "I" takes during the loop but specifies the endpoint more precisely.

*** Many Thanks Mike! ***

Ed-> This sort of article makes very good reading for 'budding' Basic programmers, especially with it's use of the Front Panel to explain the reason why problems occur.

```

10 REM *****
20 REM ** VALENTINE'S PROGRAMME **
30 REM *****
40 REM ** JOHN SEAMAN *****
50 REM *****
60 REM INITIALIZE
70 DIM HEARTS(36)
80 GENPAT 0,93,0,1,3,1,7,8,17,161
90 GENPAT 0,94,0,0,128,0,192,32,16,10
100 GENPAT 0,95,193,33,17,9,5,3,0,0
110 GENPAT 0,96,6,8,16,32,64,128,0,0
120 GENPAT 0,97,3,28,160,255,0,0,0,0
130 GENPAT 0,98,128,112,10,255,0,0,0,0
140 GENPAT 0,99,2,7,2,2,2,2,2,2
150 GENPAT 0,100,0,198,238,254,124,56,16,16
160 VS 4: COLOUR 0,15: COLOUR 2,15: COLOUR 4,15: COLOUR 1,9: COLOUR 3,
9: CLS
170 PRINT " PRESS THE SPACE BAR TO FIRE"
180 LET F=0: LET X=6: LET Y=6: LET C=1: LET Q=16
190 CSR 15,17: PRINT CHR$(93);CHR$(94)
200 CSR 15,18: PRINT CHR$(95);CHR$(96)
210 REM MOVEMENT
220 CSR X,Y: PRINT CHR$(100): PAUSE 65
230 IF INKEY$="" THEN LET F=1: CSR 15,17: PRINT " ": CSR 15,18: PRIN
T CHR$(97);CHR$(98)
240 IF F=0 THEN GOTO 290
250 CSR 15,Q: PRINT CHR$(99)
260 CSR 15,Q+1: PRINT " "
270 LET Q=Q-1
280 IF Q=Y THEN GOTO 330
290 IF X>23 OR X<6 THEN LET C=-C
300 CSR X,Y: PRINT " "
310 LET X=X+C
320 GOTO 220
330 REM ARROW FIRED
340 IF X=15 THEN GOTO 370
350 CLS : CSR 5,10: PRINT "MISSED! TRY AGAIN"
360 PAUSE 2000: GOTO 160
370 REM ON TARGET
380 CLS : CSR 13,10: PRINT "GOTCHA!": PRINT CHR$(7): PAUSE 2000: CLS
390 CSR 9,0: PRINT "BE MY VALENTINE": CSR 11,18: PRINT "SWEETHEART!"
400 FOR I=1 TO 36: READ HEARTS(I): NEXT
410 FOR I=1 TO 36 STEP 2: CSR HEARTS(I),HEARTS(I+1): PRINT CHR$(100):
PAUSE 65: NEXT
420 FOR I=1 TO 36 STEP 2
430 CSR HEARTS(I),HEARTS(I+1): PRINT " ": PAUSE 65
440 CSR HEARTS(I),HEARTS(I+1): PRINT CHR$(100)
450 NEXT
460 GOTO 420
470 DATA 15,11,16,10,17,9,18,8,19,7,19,6,18,5,17,5,16,6,15,7,14,6,13,
5,12,5,11,6,11,7,12,8,13,9,14,10

```

HARDWARE AND SOFTWARE PRICE LIST

Thanks to everyone that bought some of our 'cheapy' software, we still have plenty left, so why not go ahead and treat yourself!!! This month we can offer some of Megastar's superb new releases, all at club prices and from existing stock where possible. Review copies will be sent out to our reviewers for close inspection.

MEMORY EXPANSION BOARDS

32K £40.00
64K £50.00
128K £80.00

SPECULATOR £40.00
NEWWORD ON ROM £40.00
PASCAL ON ROM £40.00

DMX80 PRINTER £200.00
(INCL. CONN. CABLE
AND P & P)

SDX500K +I/F £250.00
SDX1MB +I/F £300.00

SDX500K +I/F
+80 COL. +CP/M
NW & SC £400.00

Software prices for the best and most popular software:-

Zarkos £6.00
Qogo2 £6.00
Surface Scanner £6.00
Chamberoids £6.00
Fathoms Deep £6.00
Quazzia £6.00
Crystal £6.00
Cee-5 £6.00
Roller Bearing £6.00
Downstream Danger £6.00
26*26 Spread Sheet £7.95
Ed/Asm £7.95
Memosketch £7.95

Dust cover's Only £3.50

!!! SUPER CHEAPIES !!! (ONLY FROM STOCK)

DESC	QTY	PRICE (Each)	DESC	QTY	PRICE (Each)
DUNGEON ADV.	2	£7.00	THE ZOO	1	£4.50
ADV. QUEST	1	£7.00	COBRA	1	£4.50
EMERALD ISLE	1	£7.00	BRIDGE	1	£4.50
MAXIMA	1	£3.50	FIRST LETTERS	1	£4.50
BLOBBO	4	£4.50	WORD & PIC MATCH	1	£4.50
KILOPEDE	1	£4.50	BASIC BUSINESS	2	£5.00
REVERSI	2	£4.50	HELI-MATHS	2	£4.00
MINEFIELD	3	£4.50	SPELLI-COPTER	2	£4.00
BACKGAMMON	1	£4.50	FIRE HOUSE FREDDIE	2	£4.00
OBLDIDS	3	£4.50	ASTROMILLON	1	£4.50
NEMO	2	£4.50			
SNAPPO	2	£4.50	FROM ELSTREE COMPUTING		
PAYROLL	1	£10.00	CUSTOMER INF FILE	1	£5.00
PURCHASE LEDGER	1	£7.00	INVOICE & CR NOTE	1	£5.00
PHYSICS 1	2	£5.50			
MATHS 1	1	£5.50			

ROM CALLS

By

Nicholas Hill

Here is a routine that uses ROM calls. If you have a hex or binary number that you want to print to the screen, it has to be converted into base 10 then ASCII first. There are ROM routines for this, which this example illustrates. The first 5 lines, clear the part of the system variables that holds the ASCII codes. The binary number to be printed is then loaded in HL (in this case, \$FF, but the value could come from elsewhere in a longer program), put into the system variable ACC, and the ROM routine called. Now the result has to be printed. HL is loaded with the start of the system variable, and then the number is printed. The routine ignores any zeros left, and the value is terminated by \$FF. The cursor has not been set, but this could be done by :-

```
RST 10
DB $83,3,X,Y
```

Main ROM Call Routine...

```
LD HL,$FDDE ;Start of area to be cleared
LD DE,$FDDF
LD (HL),0 ;Fill with zeros
LD BC,12
LDIR
LD HL,$FFFF ;Binary number to be converted
LD ($FDCB),HL ;to ASCII digits, System var
CALL $0DFE ;ACC, ROM routine
LD HL,$FDE5 ;Start of ASCII area
NN: LD DE,NUM ;Digit to be printed
LD A,(HL) ;Digit to A
CP 255 ;End of number?
RET Z
CP 0 ;or blank?
JR Z,NM
LD (DE),A
CALL PR ;Then print it
NM: INC HL ;next one...
JR NN
PR: RST 10 ;Prints
DB $81 ;One
NUM: DB $30 ;Digit
RET
```

--- Many Thanks Nicholas Hill ---

... This Is Only A Test

Types Of Problem

Programming problems can be classified in three different groups: numbers, characters and applications. Number problems contain pure mathematical questions. Generally these problems have simple inputs, but they involve complicated processes to arrive at the desired output. They are very good exercises for logical thinking.

Character problems are different from number problems. Of course they use letters instead of numbers; however, these problems normally have an easy logical structure. Like number problems, character problems can also involve fairly complicated procedures to achieve the desired results. They will test your patience as well as your programming ability.

Application programs combine characteristics of both number and character problems. They are more like the programming problems you will find in the real world. Experience is sometimes the best weapon to solve these problems.

This month and for the next couple of months, I'll leave you with a couple of programming self-tests. The first is designed to be fairly easy, taking up to about an hour to solve, the second is a bit more complicated and should take a couple of hours or more to fathom out.

Each problem will be defined and any special terms will be explained, followed by the problem itself and then input and output formats. For scoring purposes, there will be three limiting factors:-

1. Time
2. Variables used
3. Program length (Amount of program lines, one statement per line.

The table below shows the points you score for completing each exercise. Obviously, the person(s) sending in the best programs will win a prize of some sort. Answers to problems will be printed the following month in Basic and Pascal.

Problems

Exercise 1 - Calculate and Print prime numbers

Definition: A prime number is defined as any integer which can be divided only by itself and 1.

Problem: Calculate and print all the prime numbers between 2 and 100.

Input: None

Output: All the prime numbers between 2 and 100 in numerical order.

Limitations:

Time1 hour

Variables12

Program Length24 lines

Examples of prime numbers :- 3,13,37,61 etc...

Exercise 2 - Smith Numbers

Definition: Smith Numbers are a unique set of numbers in which the sum of the individual digits of the original number equals the sum of the individual digits of the prime factors whose product is the original number. (The prime numbers are included because their prime factors are themselves). For example, the sum of the digits in the number 6036 is (6+0+3+6=15). The prime factors whose product is the original number are (2*2*3*503=6036). The sum of the individual digits that comprise these prime factors is (2+2+3+5+0+3=15). The number 6036 is then by definition, a Smith Number.

Problem: Determine if a number in the range 1 through 32767, is a Smith Number.

Input: Any number between 1 and 32767. Hint, no more than 15 prime factors are needed to make up the product of an input within this range. Use the following numbers as test data; 22,5436,6036

Output: The format of the output should be a statement as to whether the input is or is not a Smith number. For example, 6036 is a Smith Number.

Limitations:

Time2 hours

Variables ...50 variables (A 15 element array counts as 15)

Prog length .60 lines

Scoring chart

Points	100	90	80	70	60	50

Exercise 1						
Time (mins)	10	20	30	40	50	60
Variables	2	4	6	8	10	12
Length	6	10	14	18	20	24
Exercise 2						
Time (Mins)	20	40	60	80	100	120
Variables	20	25	30	35	40	50
Length	30	37	45	50	55	60

Exercise 1

Time (mins) 10 20 30 40 50 60

Variables 2 4 6 8 10 12

Length 6 10 14 18 20 24

Exercise 2

Time (Mins) 20 40 60 80 100 120

Variables 20 25 30 35 40 50

Length 30 37 45 50 55 60

YOUR LETTERS

Hints & Tips

1. When the MTX is in graphics mode and ordinary text is printed the letters (characters) seem to be spaced out because they are defined for text. So to remedy this a program to make the characters "chunkier" is needed, and it just so happens that I have written such a program. Below is a short assembler code routine which:-

Reads the first byte of the character, rotates it right, stores it, rotates it left and adds the stored value, this produces "chunkier" characters for the graphics screen.

```

LD DE,£1800
LD HL,£1800
LD C,£80
LOOP1: LD B,£08
LOOP:  CALL VSET
      IN A,(1)
      RRA
      LD H,A
      RLA
      OR H
      CALL VSET1
      OUT (1),A
      INC HL
      INC DE
      DJNZ LOOP
      DEC C
      LD A,C
      JP NZ,LOOP1
      RET
VSET:  PUSH AF
      LD A,E
      OUT (2),A
      LD A,D
      OUT (2),A
      POP AF
      RET
VSET1: PUSH AF
      LD A,E
      OUT (2),A
      LD A,D
      OR £40
      OUT (2),A
      POP AF
      RET

```

Running this program "beefens-up" your characters, to try them out properly, type in this little bit of basic code:-

```

10 VS 4: CLS
20 CSR 5,5:PRINT "MEMOTECH"
30 PAUSE 5000
Thanks - Liam Redmond !!

```

2. By the way, at address £0DD0 in the ROM is a routine to put the value of the BC register (Hex) into cells £FDE5 to £FDEB. This can be examined using PANEL:-

```

LD BC,£FFFF
CALL £0DD0
CALL £2E8
RET

```

This routine will print out 65535 (£FFFF).
Thanks again to Liam Redmond !!

3. The sound routine is £8F6 not £F86 as it was printed in your October '85 edition. Liam Redmond ...Again!

4. The following Basic program is a simple 'Instring\$' emulator. It will check and keep a count of how many times 'B\$' appears in 'A\$'. However, it will only work when 'A\$' and 'B\$' are both either in UPPER or lower case:-

```

1000 CLEAR : LET A=0
1010 INPUT "String to be 'searched': ";A$
1020 INPUT "String to be looked-for: ";B$
1030 LET B=LEN (B$)
1040 LET L=LEN (A$)-B
1050 FOR I=1 TO L+1
1060 IF A$(I,B)=B$ THEN GOSUB 1100
1070 NEXT I
1080 IF A=0 THEN GOSUB 1150
1090 STOP
1100 LET A=A+1
1110 PRINT A,A$(I,B),B$
1120 RETURN
1150 PRINT "NO Match Found."
1160 RETURN

```

Best Regards - John Davidson.

5. If you're trying to save memory in a program and find yourself using a string of commands often, then DIM '\$' and assign the codes for the commands (and parameters) to this. (N.B. This will only work with codes 1-31). Then simply use :- PRINT CHND\$ <-...or any '\$' name.

P.S. How about adding a full NODDY section to the program library, with articles and/or short stories by club members? I do a bit of writing and have won a few writing competitions.

John McCormack - Co. Offaly, Eire.

Ed-> Has anyone any more ideas on this subject, or even any Noddy programs/stories?

Magazine Ideas

1. What about a joint publication, with the Swiss User Group?

Ed-> The Swiss User Group are printing some of our old articles, to help get them off the ground. I am sure they would have no objection to us printing their articles. ..Can anyone read Swiss and convert it to English for us?

Questions And Answers

1. Would it be possible to just use a (colour?) monitor with CP/M instead of both a monitor and T.V.?

Ed-> Some of the latest generation of portable 'Computer' T.V.'s have a socket for an RGB input as well as the normal T.V. input. These are suitable for use as T.V./Monitors. Switching from T.V. to Monitor only requires the pressing of the relevant channel button.

2. I have a slight problem with my MTX 512 which you might be able to help with. The computer is fitted with Newword ROM and RS 232 serial boards driven off the standard power supply unit. This all works fine when I use the unit at home but if I take it to work or to my parents home up in Southport, I consistently get bags of screen flicker (including paper colour), horizontal dark bands down the screen and such like. This occurs within a couple of minutes of power up. The last time it did this I removed the ROM and serial boards and this cleared the trouble.

I suspect that the above problem is due to lack of power available from the PSU when the mains voltage is on the low side and both boards (taking up a bit more power) are present.

Ed-> Your problem is most certainly a lack of power, some areas of the country suffer from low voltage (around 220volts instead of 240volts), this especially happens around peak power consumption times. Memotech will probably solve this problem for you, (if you ask them nicely!), I think they have an alternative power supply unit.

3. I have a small problem with my in-board Newword ROM 2; every now and again on boot-up ROM 2 drops straight through to Basic or Panel, without being available. Any idea why and how I can cure it. At the moment I open the machine and press the Newword board chips and connections. This seems to cure it for a while. Any more permanent cure?

Ed-> Undoubtably your problem is a bad connection between your main board and your ROM board. Try disconnecting the ROM board, cleaning up the connections on both boards, with very light emery cloth and connecting back together.

4. I have a query concerning video ram. As you know when using sprites in games programming it is very useful to know the positions of sprites which roam around the screen independant of program control. Other computers which use th T.I. chip provide the facility to read the positions of sprites ie the TI99/4A uses CALL POSITION(Sprite No,X,Y) or VPEEK, and MSX machines also have the ability to VPEEK.

Ed->The Sprite Attribute table contains the necessary info, (see issue 6 vol 1), you will see that each sprite has 4 bytes, the first two of which hold the X,Y position of the sprite. It will be necessary to set the VDP to read mode then access the corresponding bytes for the sprites from the attribute table. The attribute table starts at 6144 (Dec).

Does anyone have a routine that will do this!!!.

Ideas

1. I found a project that may be of interest to Memotech users in the February 1986 issue of Everyday Electronics & Electronics Monthly. It is a touch controller, an electronic device which uses the electrical resistance of the operators skin to activate electronic switches, that replaces the joystick. It plugs into the joystick ports and is said to be compatible with all joystick software. If you wish, I could supply a report on it's use and reliability once I have constructed it.

Mr K.Wood Gosport, Hampshire

Ed-> We look forward to hearing more about this in the future!!!

Adventure Help Lines

1. I would be grateful if you could help me with a problem I have with one of my Continental Software games especially "Alice" and that is, how on earth does one get past the guard Dragons and also the Cheshire cat which blocks the door into the Duchesses house.

INTERFACING IDEAS

About a month ago, I had another one of those re-occurring urges to built an electronics device for my MTX. This time it's an A/D converter. Why?, well I had this idea to make a simple oscilloscope, showing real time conversions on the MTX graphics screen. It could of course be used for any type of analogue input device, ie temperature sensor, photo-cell (light sensor) etc. A few weeks ground work uncovered a lot of data about D/A converters, they can really be broken down into three main catagories, Simple (cheap, slow and inaccurate!), Expensive (Very quick and accurate, used for specialised applications) and "About as Much as the Pocket can Afford!", (This seems to be the right one!).

More about the "About as Much as the Pocket can Afford" type of A/D converter. Mostly home electronics are built around the size of your pocket (mine ain't very big), A/D converters are no different, I'm sure if I could afford the £30-£50 for a high speed 'flash' converter I would buy one, but I can't, so these cheaper one's (up to about 5 quid) are a compromise, they are fairly fast at up to 66000 conversions a second, also inputs are sufficiently accurate for home use and they are simple to use.

However, in order to have one that's reasonable, 8 input data lines are needed and 3 more input/output lines for 'chip' control, this means that using the internal port is out of the question, as it is just not powerful enough. This is a shame as up to now this has sufficed for all my interfacing needs. The 'ports' problem can be overcome by building an interfacing board to fit onto the edge connector. There are two 'chips' on the market that will create the required data ports:-

1. The Z80 PIO Chip, this is part of the dedicated Z80 family of chips and would appear the obvious choose. It creates two input/output ports, and a small amount of control.
2. The 8255 Chip, this is a less obvious choose, but if you look at it's specification a little closer, it appears to be slightly better, at a slightly lower price. It offers 24 programmable input/output lines, which are configurable in many different ways. As you may have guessed this is the one that I have 'plumped' for.

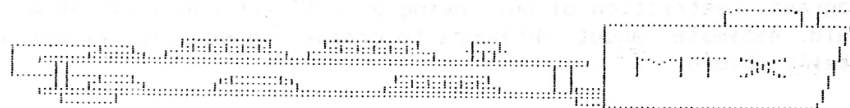
As yet I have 8 bits of this port chip working, and am working on the final conections for the remaining lines. The whole board has proved remarkably simple in design and build, it consists of only two chips (in sockets), the rest is done with wire conections.

The port board fits easily on a 4" x 5" piece of vero board, in order to save space the 'applications' boards will fit 'piggy' back fashion on top of the port board, (as in the sketch below) this will also help keep things rigid.

Another little 'gimmick' I've thought of including, is a set of LED's for one of the ports, this would give a good visual effect for those not in the 'know' about computers, convincing them that this fantastic board of electronics is infact doing something!.

When this is all built and tested (note of optimism!!), it should be really interesting to make a program that actively uses the converter in real time, displaying input graphically on the MTX graphics screen.

Anyone wanting details can find most of the necessary info in a book called Micro Interfacing Circuits Book 1, by R.A. Penfold. The publisher is Bernard Babani (BP130). It only costs £2.25!! Technical Data sheets are available for all Integrated Circuits from most suppliers, for a small fee. (Usually 50p).



Reviews... Reviews... Reviews... Reviews... Reviews... Reviews...

SOFTWARE REVIEWS

Roller Bearing

Publisher - Megastar

Price - £6.50

Battle against maddening magnets and acid dust clouds said the instructions.

So I loaded the game, up flicked a neat loading screen but to my horror those 'orrible lines like what the plectrum! sorry Spectrum uses when it's loading appeared, (how degrading) but to my relief after a couple of minutes the game had loaded and a neat menu flicked onto the screen, with an absolutely 'brill' version of the 'Brookside' theme tune, the sound really was amazing.

The game was also quite a surprise with nice large colourful graphics, but I must say that the game is very difficult and the Bearing takes quite some controlling.

The idea of the game is to get past various nasties, hills and bumps to get to a waving flag.

I must say thanks to Mr Butterfield this must become a number one for the MTX.

Sound - 10

Lasting Interest - 8

Graphics - 8

Value - 8

Reviewed by Richard Siddall

Memosketch

Publisher - Syntaxsoft

Price - £7.95

The first sentence of the Memosketch manual reads :-

"Memosketch is a utility package that presents the ultimate in MTX software."

Memosketch's design is very good, you can view the whole graphics screen or zoom in to one particular section. You can zoom up to a magnification of *8, which makes one pixel 8*8 square. From a menu you can change ink and paper colours but you are still restricted by the VDP's physical ability to only display one ink and paper colour per 8 columns.

The program comes with a couple of ready drawn examples which are really impressive and will match the quality of anything around. It is possible to screen dump any picture, screen colours are shaded tones of grey on the print-out, which fits nicely on an A4 sheet.

Unfortunately, it takes 'a hell of a long time' to create a picture, due to the programs restriction of only being able to set one pixel at a time. Infact I would estimate about 40 hours to create a good picture, which I feel to be totally absurd.

Sound N/A

Lasting Interest 8

Graphics 6 (Ease of drawing)

Value 8

PROGRAM LIBRARY
£1 Per Cassette, 2 Programs per Cassette

This month we have two new programs, the first is a Swiss program (sent in by a non-member), it is called Mice (Mauce), it is written in Basic and is rather good and would really suit anyone as it is very simple to play but a real challenge never the less. The second program was sent in by Mr P.Wood, it is another arcade game, written entirely in assembler, a feat in itself worth seeing. This one is a must for anyone's library of useful programs. Christian Woehlbier of the Swiss User Group sent me a diskette full of programs many of which were in Basic, more important in terms of worth of program was a Small C Compiler. Unfortunately, The diskette was 'zapped' by the X-Ray machines, so I've had to send it back to get it re-copied.

1. Basic & Assembler Programs

All programs available on cassette, 2 programs per cassette, £1 per cassette. Or on disc, £2.50 per disc, please enclose a disc, stating capacity. (Some programs are only available on cassette!!).

- 1.Hex-Dec-Bin Conversions. (Binary Bit In Assembler)
- 2.CGEN Sprite Generator.
- 3.3D Drawing Board. Rotate a skeleton of a cup & saucer in 3D.
- 4.Whist. The Card Game
- 5.Memory Save. This Utility will Save a block of memory to tape and retrieve it.
- 6.MTX Drawing B'rd. ;Two basic drawing boards, MTX DB has
- 7.LUGO Drawing B'rd.;more extensive commands
- 8.Simplex Tablaeux. Applications Program
- 9.Breakeven. Applications Program
- 10.Statistics Applications Program
- 11.An Unsolved Prbm Applications Program
- 12.Radio Routines Applications Program
- 13.Light Cycles. Arcade Game
- 14.Hex/Dec/Bin Conversions using USER commands!
- 15.Renumber II Renumbers Including GOTO's etc
(14 & 15) are Utilities and as such reside high in memory transparent to the user.
- 16.RELOC ; Relocs Assembler Properly!!
- 17.Character; Editor Yepp!! Another Sprite Gen!!
- 18.Quasimodo; Excellent Arcade Game
- 19.Planner ; YASG (Yet Another Sprite Generator)
- 20.Hanoi ; Classic Puzzle (Brilliant simple use of
- 21.Noble ; Simple Text Game Graphics)
- 22.Hi-Lo ; Just like Bruce's Play Your Cards Right
- 23.Composer ; Our First Sound Generator!!
- 24.Anova ; Applications Program
- 25.CASHFLOW; Applications Program
- 26.RenuaIII ; Utility !!!26,27 & 28 cassette only!!!
- 27.Merge ; Utility
- 28.Money Manager ; Applications program
- 29.Word ; Word Processor
- 30.Reversi ; Strategy Board Game
- 31.Full Time; Football Manager Game
- 32.PANEL3 ; Panel extensions
- 33.Texted ; Word-pro
- 34.Mice ; *** New Swiss Arcade game Written in Basic
- 35.TNTTIM ; *** New *** Assembler arcade game.

2. Programs/Procedures in Pascal

(Available as listings or on disc. Please provide sufficient postage to cover club costs!!)

1. DBASE for Disc Turbo Pascal
- 1(a). Comprehensive Create File Procedure
- 1(b). Simple Display File Procedure
- 1(c). Add More Data To File

3. Articles From Previous Magazines

(Available as listings, please provide sufficient postage to cover club costs. TA!)

- 1.PANEL2 Utility. An updated version of PANEL1, which includes a second feature.
- 2.Undocumented Neword dot commands.(Vol1 Iss.7)
- 3.Hisoft Pascal Review (vol1 Iss.8)
- 4.Neword Rom Review (Vol1 Iss.5)
- 5.RST10 Codes Explained (Vol1 Iss.3)
- 6.VDP Explained Using assembler (vol1 Iss4,5,6)
- 7.System Variables (Not Previously Published!!)

4. CP/M Programs/Utilities

(!!! Available only on disc !!!, please send in a formatted disc stating capacity and enough postage to cover).

- 1.A simple mail label system for up to 3 across labels, written in EBasic. Disc includes Ebasic compiler and runtime program. Consists of a suite of half a dozen programs. (Ensure that you send in at least 250K of blank disc!!)
- 2.PLOT33 A new graphics plotting package for Turbo Pascal owners. Create and print your own graphics. Set up for DMX type printers but will support most others. Must be seen to be believed. Please ensure you have at least two weeks free when ordering this one, you'll need it!!. (Ensure that you send in at least 300K of blank disc for this one).
3. Z80.ASM This is a Z80 assembler to replace the ordinary CP/M assembler which uses the 8080 mnemonic command set. Z80.ASM supports all the features of the notable Ed/Asm, especially macro libraries and a slightly more standard Z80 mnemonic command set. The disc also contains a Z8 assembler. Please send in a formatted 500K disc to ensure there will be sufficient space for all the programs.

This file was downloaded from

www.primrosebank.net

Abridged Terms & Conditions (Downloads)

Disclaimer

www.primrosebank.net, (*the website*) is provided by Dave Stevenson as a service to the public, is provided "as is" and carries no warranties, expressed or implied, of any kind.

Dave Stevenson is not responsible for, and expressly disclaims all liability for, damages of any kind arising out of use, reference to, or reliance on any information contained within the website or made available for download. Whilst the information contained within the website site is periodically updated, no guarantee is given that the information provided on the website is correct, complete, and up-to-date.

A number of articles on the website contain technical data and practical guidance which may be of use in testing and maintaining various items of vintage computer and electronics hardware. Such articles are not intended to cover all aspects of the tasks involved and may omit essential information, including necessary safety precautions. Performance of the tasks described may risk damage to equipment and/or people. The reader is responsible for ensuring that he/she is capable of performing the tasks described and well as assessing the inherent risks involved and taking appropriate measures to mitigate such risks.

Dave Stevenson expressly disclaims all liability for, damages to equipment or injury of any kind arising out of use of such technical data and guidance.

Unless otherwise noted, all data on the website is deemed to be *Copyright (c) Dave Stevenson, 2009-2013*

You are hereby granted permission to download data and software from the website *for your own personal use. Redistribution of any content from the website without written authorisation from Dave Stevenson is expressly forbidden. You are also expressly forbidden from offering for sale any material obtained from the website.*

As far as possible, information included on the website from other sources has been credited to the respective author and/or publisher. The majority of content on the website is derived from material first published in the 1980s. *This material is likely still under copyright of the original author and/or publishers.* The authors and/or publishers may not have given express permission to copy, transmit or make this information available for download, but I believe that they would have no objection to this archive information being placed into the public domain.

However, should the author and/or publisher of the original material find any content on the website for which they wish to assert their rights, they should notify Dave Stevenson (by e-mail to: webmaster@primrosebank.net) who would be pleased to enter into a dialogue to agree a satisfactory resolution of their concerns.

If you obtained this file as part of a paid-for package, you have been scammed! I suggest that you request a refund from the seller, please also advise Dave Stevenson at the e-mail address above.