

1st TECHNICAL REPORT 1984

Since the release of the MTX series of Microcomputers There have been several Questions posed that were initially unanswerable. In this report we hope to provide some answers to these questions and some additional Information not contained in early Versions of the manual.

The Report is Broken down into 2 sections.

SECTION 1

Questions That our customers have asked.

CHAPTER	PAGE
1.SOUND	
2.VIDEO	
3.BASIC	
4.NODDY	
5.PANEL & ASSEMBLER	
6.CASSETTE	
7.PRINTER	
8.HARDWARE	
9.FDX DISK UNIT	
10.KEYBOARD	

SECTION 2

MACHINE CODE AND THE MTX.

CHAPTER	PAGE
1.INTRODUCTION	1
2.VIDEO	2
3.SOUND	
4.KEYBOARD	
5.ARITHMETIC	

MACHINE CODE AND THE MTX 1.INTRODUCTION

The MTX Series of Micro Computers is Z80 based and as a result is very easy to program in machine code. This is not designed as a Tutorial in machine code programming but nearly gives some assistance in the non standard points of the MTX, and how to best make use of them.

It is important to have a good knowledge of the manual and of the MTX assembler before attempting to use the routines and functions presented in this machine code aid. Particular attention is drawn to the SYSTEM variables in SOFTWARE APPENDIX 5 and the comments made after it.

If a machine code call to the MTX ROM requires a result then the location where the result can be found will be put in brackets afterwards. eg. GR\$ returns it's result in WKAREA which is at #FE1A.

LD A_j(LOOP²)

MACHINE CODE AND THE MTX 2.VIDEO

In an aid to overcome any complications with using the TMS 9918 series Video Processor All the functions are easily available using the screen restart (RST 10). With various different applications, in general the screen restart sends a group of characters or Control characters to the device specified by the IOPR system variable (see P 182 of your manual)

The Screen Restart is operated by calling RST 10. This should be followed by a series of commands and DATA. The Commands can be separated into four different categories.

1.OUTPUT BC TO THE IOPR DEVICE

Format:

Print char c; char B.

7	6	5	4	3	2	1	0

1	1	C	-	-	-	-	-

BC TO SCREEN

BIT C

Bit C is 0 when you wish no more bytes to be sent to the IOPR device, When set to 1 the Screen restart routine checks subsequent bytes to be sent until bit C is found to be 0.

USAGE

eg.

```
BCOUT:  RST 10
        DB 192 ; Sends the contents of BC to screen.
        RET
```

2.OUTPUT N BYTES TO THE IOPR DEVICE

Format:

7	6	5	4	3	2	1	0	

1	0	C	<---N---	>				

send 'N' bytes to IOPR DEVICE

BIT C as above

This is possibly the most frequently used screen restart call. It allows N bytes (following the RST 10 call) to be sent to the screen (Max 31 or #1F). As with all uses of the Screen Restart it is best to use a DB command in the assembler with the N bytes in it.

USAGE

eg.

```
VDOUT:  RST 10
        DB #8E, "*MEMOTECH LTD*"; Send MEMOTECH LTD to
screen.
        RET
```

MACHINE CODE AND THE MTX

2.VIDEO (Contd.)

3.SELECT AND CLEAR VIRTUAL SCREENS

Format:

```

  7  6  5  4  3  2  1  0
  -----
  0  1  C  -  cls <--N-->
  -----

```

BIT C as above

BIT 4 is 'Don't care'

BIT 3:CLS

The CLS Bit when set will cause the virtual screen (N) to clear on entry and is equivalent in MTX Basic to VS n:CLS.

When the CLS Bit is not set entry is made to the Virtual Screen (Defined by N) Leaving the Contents of the screen untouched.

USAGE

eg.

```

VSSEL:   RST 10
         DB 4C ; Call VS 4 and Clear it on entry.
         RET

```

4.SEND ONE BYTE TO THE IOPR DEVICE

Format:

```

  7  6  5  4  3  2  1  0
  -----
  0  0  C  <-----N----->
  -----

```

BIT C as above

This call to the screen restart allows a one byte command to be sent to the IOPR device. This may be a Clear Screen, Control Code or some type of printer specific command (assuming that IOPR is set to the correct value (1) for data to be passed to the printer). One other method of sending a single byte to the screen is to load the byte into the A register and CALL #BC. This is totally independant of the screen restart.

MACHINE CODE AND THE MTX

2.VIDEO (Contd.)

SOME EXAMPLES OF THE SCREEN RESTART

```

0001      RST 10                      ;CALL SCREEN RESTART
0002      DB #64,#AA,10,10,"MEMOTECH" ;SELECT VS 4, LINEFEED TWICE
                                AND PRINT 'MEMOTECH'
000E      DB #88,26,"THIS IS"        ;HOME CURSOR AND PRINT 'THIS
                                IS'

```

In line 2 the #64 selects VS 4 and as the BIT C is set (BIT 5) the restart routine carries on to the next Byte. #AA has bit 5 set so that the process will continue on to 000E, and causes the next 10 bytes to be output to the Screen (assuming that IOPR is set to 0). The 2 Ten's are the codes for two Line feeds.

In 000E you may note that #88 does not have bit 5 set, thus this will be the last command executed by the screen restart. The 26 causes a home cursor so that 'this is' is output above the 'Memotech' displayed.

MACHINE CODE AND THE MTX

2.VIDEO (Contd.)

USING SCREEN RESTART

This section of the report is designed to show how you can use the screen restart to implement the graphics calls in the MTX Rom. An example is shown below of the method used in general and following this section there are some examples of the use of this method.

```
RST 10          ;Call screen restart
DB <Call to output N chars>,<Command>,<Params>
```

ASCII CODE	CTRL/ESC SEQUENCE	FUNCTION
1	CTRL A	PLOT X,Y
2	CTRL B	LINE X1,Y1,X2,Y2
3	CTRL C	CURSOR X,Y
14	CTRL N	CTLSPR P,X
15	CTRL O	GENPAT P,N,D1,D2,D3,D4,D5,D6,D7,D8
16	CTRL P	COLOUR P,N
17	CTRL Q	ADJSPR P,N,V
18	CTRL R	SPRITE N,P,XP,YP,XS,YS,COL
19	CTRL S	MOVSPR P,N,D
20	CTRL T	VIEW DIR,DIS
21	CTRL U	INSERT KEY
22	CTRL V	DELETE KEY
23	CTRL W	BACK TAB
25	CTRL Y	TAB KEY
26	CTRL Z	HOME KEY
27,65	ESC 'A'	ATTR P,STATE
27,89	ESC 'Y'	CRVS N,T,X,Y,W,H,S
27,90	ESC 'Z'	VS N
27,67	ESC 'C'	GR# X,Y,B (RESULT IN WKAREA)

The above routines are mainly command words except for the ones marked KEY and CTRL W. These we have included as they are also very useful routines to include in your display sections.

SOME EXAMPLES OF USING THE ROM CALLS ON SCREEN RESTART

```
RST 10          ;SCREEN RESTART
DB #64,#A5,2,0,0,0,192 ;LINE 0,0,0,192
DB #A5,2,0,192,255,192 ;LINE 0,192,255,192
DB #A5,2,255,192,255,0 ;LINE 255,192,255,192
DB #A5,2,255,0,0,0 ;LINE 255,192,0,0
DB #9C,3,10,10,"THIS IS A SQUARE"
RET            ;END OF ROUTINE
```

```
OR
RST 10
DB #64,#B4,2,0,0,0,192,2,0,192,255,192,2,255,192,255,0,2,255,0,0,0
DB #9C,3,10,10,"THIS IS A SQUARE"
```

MACHINE CODE AND THE MTX

2.VIDEO (Contd.)

Both of the above routines are in effect the same and give the same results. As you can see from the remarks made at the side of the lines, the routine will draw a square around the edge of the screen, and then print THIS IS A SQUARE at the point 10,10. It uses the CTRL B(line) and CTRL C(csr) functions.

MACHINE CODE AND THE MTX

3. SOUND

MACHINE CODE AND THE MTX 4.KEYBOARD

As described in the PORT summary in the manual, the keyboard can be read as an 8 x 10 Matrix using port 5 to output 8 drive lines and using ports 5 & 6 to read the sense lines. This however is not necessary as there is a ROM call which is equivalent to the INKEY\$ command in BASIC except for debounce checking which does not occur in the BASIC form of INKEY\$.

A call should be made to #79 and the result can be found in the A register. If the result is either a null character (CHR\$(0)) or the character is the same as the previous one typed (and there has been no change since the previously typed character) then the ZERO flag will be set, so if debounce is not desired then a 'JR z' would be necessary.

MACHINE CODE AND THE MTX

5.ARITHMETIC