

Interesting graphic routines are easily available to MTX machine code programmers, says Keith Hook.

# Mastering MTX code

The Memotech has one of the most sophisticated graphic command sets available to the Basic programmer. My preference for programming in Assembler has always made me envious of the way you can manipulate sprites, draw lines and animate graphics in Basic, but after studying a print-out of the ROM and making a few phone calls to Memotech, the truth of the matter dawned on me. The ROM is so well thought out that the majority of graphic routines can be available to the machine code programmer, and are as easily implemented as they are in Basic.

The key lies with the RST10 instruction. An RST instruction is a unique 1 byte command that allows a call to any one of eight addresses in low memory. And as it requires only 1 byte to implement, it's a very fast instruction.

RST 10 is used by the MTX for at least 90 per cent of ROM graphic routines available from Basic. The MTX ROM has been written so that routines can be called by the machine code programmer from within a program just as easily as from Basic. The routines here should help machine code programmers.

The function of the RST10 instruction is to send ASCII or control characters to the screen or printer depending on which bit is set in the system variable 10PR. This article assumes that all writing will take place to the screen.

There are four command codes in which the screen restart (RST 10) can operate:

- Send X number of characters to screen.
- Send one byte to screen.
- Clear and select virtual screen.
- Output contents of BC registers to screen.

On calling RST 10, the screen restart expects to find a series of data or command bytes. The formula for these bytes is *command, data*.

The screen restart decides which command is to be executed by the bit pattern of the first byte, which is the command byte. The fifth bit of each command byte is always the CS bit (CarryonStop bit). With this bit set, and the immediate command executed, the screen restart carries on to look for another command byte. If the CS bit is reset (0), the screen restart stops after carrying out the immediate command byte.

## Clear and select virtual screen

Bit	7	6	5	4	3	2	1	0
Pattern	0	1	CS	—	Cls	VS	Number	

- Bit 7 is reset (0)
- Bit 6 is set (1)
- Bit 5 is CarryonStop bit
- Bit 4 doesn't matter

- Bit 3 If set (1) clears virtual screen. If reset (0) will not clear screen.

- Bits 2 to 0 specify virtual screen number.

Example: If you wish to select virtual screen 4, print 'PCN'. The formula is:

```
RST 10
DB 64,83,"PCN"
```

If you run this short programme you will see PCN printed on virtual screen 4. This works as follows:

Bit	7	6	5	4	3	2	1	0
Pattern	0	1	1	0	0	1	0	0

= 64Hex

Here bits 7 and 6 comply with the select virtual screen command. Bit 5 is set, so after selecting VS4 the screen restart will carry on and look for another command byte. Bits 2 to 0 = 4 decimal and therefore the screen restart selects virtual screen 4. The next command byte is 83Hex, which tells the screen restart to send 3 bytes to the screen and as 83Hex does not have bit 5 set the RST 10 instruction stops after printing 'PCN'.

Should you wish to select virtual screen 4 and do nothing else the formula would be:

```
RST 10
DB 44H,
44Hex = 01000100, and as bit 5 is not set,
the command stops after selecting virtual
screen 4.
```

To select virtual screen 4 and also clear it on entry before printing 'PCN' the program would be:

```
RST 10
DB 6CH,83H,"PCN"
Bit 5 is now set, as is bit 3. 6CH = 01101100
= Select VS4:CLS:Print PCN
```

## Send X number of bytes to screen

Bit	7	6	5	4	3	2	1	0
Pattern	1	0	CS	(no bytes .....				

Maximum value of X = 31  
This command is recognised by RST 10 when bit 7 is set (1) and bit 6 is reset (0). Bit 5 is the CarryonStop bit, and bits 4 to 0 specify the number of bytes to a maximum of 31. Study the above and you will see that if you wish to output 5 bytes to the screen the formula is 05 ADD 80H = 85H = 10000101. If you wanted to print 5 bytes and carry on to the next command, the formula would be 5 ADD A0H = A5H = 10100101. For example:

Print "GO IN PEACE" TO VS4:

```
RST 10
DB 06CH; Select VS4 and carry on (bit
5 set)
DB 8BH; 11 characters including
spaces
DB "GO IN PEACE"
```

Obviously you can write it all on one line in actual programming:

```
RST 10
DB 06CH,8BH,"GO IN PEACE"
```

For example:

```
Print "GO IN PEACE":CLS:PRINT "THE
END":
RST 10
DB 0ABH; Set bit 5 (carryon bit) + 11
characters
DB "GO IN PEACE"
DB 06CH; Select VS4 and set bit 5
DB "GO IN PEACE"
DB 87H; Print 7 characters but stop
after this so reset bit 5
DB "THEEND"; 6 characters + 1 space
```

The graphics with output X number of bytes to screen is the most useful of the screen restarts, and the bit pattern is exactly the same as the above instruction.

To understand how this mode works, see table 1. The parameters listed are exactly as printed in the Basic instruction manual. If you study the ASCII command digits, you will see that they correspond to the ASCII CTRL characters used to control the cursor, etc.

If one of these ASCII control digits follows the command byte the screen restart performs the function specified in table 1. Note that the value of the number of bytes must include the control character.

The format is therefore output X characters, ASCII command, parameters. For example:

```
RST 10; call screen restart
DB 6CH; select and clear virtual screen 4 set
bit 5 carry on
DB 83H; output 3 characters to screen and
stop bit 5 reset
DB 1,0AH,0AH; ASCII 1 = Plot = PLOT
10, 10
```

or

```
RST 10
DB 6C,0A3H,3,2,2
DB 8EH,"THIS IS A DEMO"
```

The above code selects and clears virtual screen 4 then outputs 3 bytes to the screen.

To test your grasp of the rules, try to sort out why this does exactly the same thing:

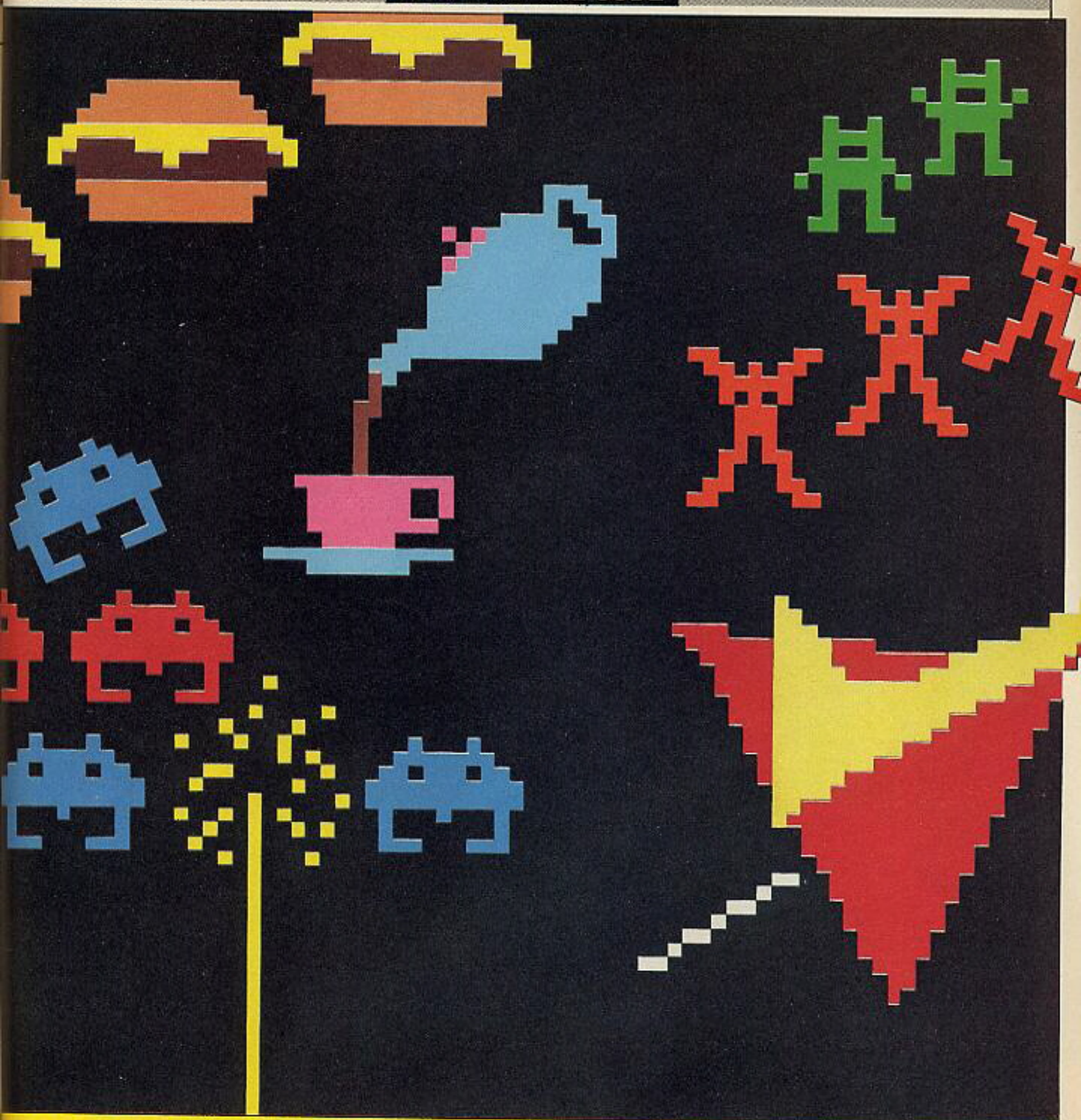
```
RST 10
DB 6CH,91H,3,2,2,"THIS IS A DEMO"
If you wanted to create a sprite the code
would be:
RST 10
DB 0ABH, call to output 11 bytes
0FH,1,1, ASCII command 15 = GENPAT
64,16,16,32,128,64,128,08 parameters
```

## Send one byte to screen

Bit	7	6	5	4	3	2	1	0
Pattern	0	0	CS	(byte .....				

The screen restart recognises this command when both bit 6 and bit 7 are reset (0). In this mode a one byte control or command such as CLS can be directed to the screen.





## Output the BC register pair to the screen

Bit	7	6	5	4	3	2	1	0
Pattern	1	1	CS	—	—	—	—	—

This mode is entered when bits 6 and 7 are set (1). Bit 5 is the CarryonStop bit, Bits 4 to 0 do not matter.

For example: LD B, 41H; ASCII A  
LD C, 44H; ASCII D  
RST 10; Call screen restart  
DB 0C0H; Output BC to screen then stop. Bit 5 = 0  
LOOP: JR LOOP; Loop to preserve screen display

This will print the ASCII characters AD to the screen.

PCN

### TABLE 1

ASCII Command Code	Function I	Parameters I
1	PLOT	X,Y
2	LINE	X1,Y1,X2,Y2
3	CURSOR	X,Y
14	CTLSPR	P,X
15	GENPAT	P,N,D1,D2,D3,D4,D5,D6,D7,D8
16	COLOUR	P,N
17	ADJSPR	P,N,V
18	SPRITE	N,P,XP,YP,XS,YS, COL
19	MOVSPR	P,N,D
20	VIEW	DIR,DIS
21	INSERT	
22	DELETE	
23	BACK TAB	
25	TAB	
26	HOME	

Formula: DB (X),(ASCII Command), (Parameters) where X is the call to output X number of characters.  
X = (ASCII Command) + (Parameters)

### Table 2

VRAM locations used by ROM

Text Mode:	
Pattern Name Table (Screen)	1C00 Hex
Pattern Generator (Graphics)	1800 Hex
Graphic Mode 2:	
Pattern Name Table (Screen)	3C00 Hex
Pattern Generator (Graphics)	0000 Hex to 17FF Hex
Colour Table	2000 Hex to 37FF Hex
Sprite Attribute Table	3F00 Hex
Sprite Generator Table	3800 Hex