



DRAGON DATA LIMITED

Dragon Data Ltd Kenfig Industrial Estate Margam Port Talbot West Glamorgan SA13 2PE
Tel: Kenfig Hill (0656) 744700 Telex: 498934

IR/JML

Dear Sir/Madam,

MACHINE CODE STARTER PACK

Thank you for buying our exclusive Machine Code Starter Pack. Included in the pack, as you know, are various routines that enable you to improve your programming by using Machine Code.

To clarify a few points, the letter that refers to AUTO-RUNNING a program should be tied in with four A4 sheets titled RUN-ON-RESET, DISABLE BREAK and LOAD WITHOUT HEADER. The only sheet untitled has three-quarters of mnemonics on it starting with ORG 200, and should be sheet 1 titled AUTO-RUN.

I hope this special offer gives you a great deal of enjoyment and allows you to use your Dragon's "hidden" facilities to their fullest capacity.

Yours faithfully,

KJ. Sale

Marketing Department

Enc.

DRAGON DATA LIMITED

Dragon Data Ltd Kenfig Industrial Estate Margam Port Talbot West Glamorgan SA13 2PE
Tel: Kenfig Hill (0656) 744700 Telex: 498934

Dear ,

Please find enclosed all the information you will require for auto start and a disabled break key.

There now follows a step by step description of how to use this information.

1. Turn on your machine
2. Load part 1 in at address &H200 onwards by either POKEing or using a hexadecimal loader
3. After part 1 has loaded, load part 2 by using the POKES described:
4. POKE &HFF03, (PEEK(&HFF03)AND &HFE)
POKE &H10D,2
POKE &H10E,0
5. To save the auto start:-
CASVEM"AUTO",&H10D,&H24C,&H200
6. Do not rewind the tape
7. Turn the machine off and on again, then load your basic program.
8. Load part 3 by POKEing into address &H200 onwards
9. Put the remote into the cassette recorder and press the PLAY and RECORD buttons.

Cont/

10. Type EXEC &H200 (ENTER)

The basic program should now be saved directly after the auto start in the format required by the auto start.

I hope that this information is of assistance to you, but should you have any further queries, do not hesitate to contact this department again.

Yours sincerely,

Encs.

0200	0200		ORG	\$200
0200	B6FF03		LDA	\$FF03
0203	84FE		ANDA	#\$FE
0205	B7FF03		STA	\$FF03
0208	CC9D3D		LDD	#\$9D3D
020B	FD010D		STD	\$10D
020E	B6FF03		LDA	\$FF03
0211	8A01		ORA	#1
0213	B7FF03		STA	\$FF03
0216	7F1E00		CLR	\$1E00
0219	BE1E01		LDX	#\$1E01
021C	9F19		STX	\$19
021E	BDB75B		JSR	\$B75B
0221	BD83ED		JSR	\$83ED
0224	3002		LEAX	2,X
0226	9F1B		STX	\$1B
0228	9F1D		STX	\$1D
022A	9F1F		STX	\$1F
022C	12	RESET	NOP	
022D	B6FF03		LDA	\$FF03
0230	8A01		ORA	#1
0232	B7FF03		STA	\$FF03
0235	8655		LDA	#\$55
0237	9771		STA	\$71
0239	308DF0		LEAX	RESET, PCR
023C	9F72		STX	\$72
023E	308C06		LEAX	LABEL, PCR
0241	3263		LEAS	3, S
0243	4F		CLRA	
0244	7E837D		JMP	\$837D
0247	2052554E20	LABEL	FCC/ RUN/	
024C	00		FCB	0
024D				

2

To disable BREAK, type the following commands at the beginning of a program.

Disable BREAK

POKE 411,228
POKE 412,203
POKE 413,4
POKE 414,237
POKE 415,228

Disable BREAK - POKE 410,236
Enable BREAK - POKE 410,57

```
4000          *****RUN ON RESET
4000 12      RESET NOP
4001 8E4000      LDX #RESET
4004 9F72        STX #72
4006 308C06      LEAX LABEL,PCR
4009 3263        LEAS 3,S
400B 4F          CLRA
400C 7E837D      JMP #837D
400F 2052554E20 LABEL FCC/ RUN ./
4015 00          FCB 0
4016
4016
```

* LOAD WITHOUT HEADER
* LDX WITH START ADDRESS
JSR \$B75B
RTS

4000 * BASIC SAVE WITHOUT HEADER
4000 0F7C CLR \$7C
4002 86FF LDA #\$FF
4004 977D STA \$7D
4006 8602 LDA #2
4008 9790 STA 144
400A 4F CLRA
400E BDB6AB JSR \$B6AB
400E 39 RTS
400F

8E0401		LDX #\$401
A680	LOOP	LDA ,X+
A71E		STA -2,X
1F10		TFR X,D
C41F		ANDB #\$1F
C11F		CMPB #\$1F
26F4		BNE LOOP
C68F		LDB #143
A681		LDA ,X++
E71E		STB -2,X
A71D		STA -3,X
8C0600		CMPX #\$600
2FE7		BLE LOOP
39		RTS

SCROLL RIGHT

8E041F		LDX #\$41F
A682	LOOP	LDA ,-X
A701		STA 1,X
1F10		TFR X,D
C41F		ANDB #\$1F
C100		CMPB #\$0
26F4		BNE LOOP
C68F		LDB #143
E784		STB ,X
30883F		LEAX 63,X
8C0600		CMPX #\$600
2FE8		BLE LOOP
39		RTS

SCROLL UP

8E0420		LDX #\$420
108E0400		LDY #\$400/
A680	LOOP	LDA ,X+
A7A0		STA ,Y+
8C0600		CMPX #\$600
2FF7		BLE LOOP
8E05E0		LDX #\$5E0
868F		LDA #143
A780	LOOP2	STA ,X+
8C0600/		CMPX #\$600
2FF9		BLE LOOP2
39		RTS

SCROLL DOWN

8E05DF		LDX #\$5DF
108E05FF		LDY #\$5FF
A682	LOOP	LDA , -X
A7A2		STA , -Y
8C0400		CMPX #\$400
2CF7		BGE LOOP
8E0400		LDX #\$400
868F		LDA #143
A780	LOOP2	STA , X+
8C041F		CMPX #\$41F
2FF9		BLE LOOP2
39		RTS

HOW TO ACCESS HIGH RES MODES IN

MACHINE CODE

There are three main areas in the Dragon which must be written to obtain graphics. These are:

1. &HFF22 - Video Display Generator
2. &FFC0 - SAM (Vertical Resolution).
3. &HFFC6 - &HFFD3 - SAM Chip (Video Display Offset).

The most important area being the Video Display Generator.

I have enclosed details on the location FF22 giving references on bits 3 - 7 of this byte.

The CSS bit controls the screen type, i.e., if CSS = 1 then you obtain a screen ?,1 is clear, screen ?,0.

GM0 - GM2 are all resolution bits setting all three to 1 will give a horizontal resolution of 256 pixels (PMODE 4). The most important of these three is GM0/I/E. This, when set, will give a two colour set and when clear will give a four colour set.

The last bit (bit 7 A/G) is the bit which controls the Alphanumeric (text) or graphic mode. Setting this bit to 1 will give graphics.

The SAM chip controls both the vertical resolution and the display offset. Locations in the SAM are arranged in pairs, the even location of the pair is used to clear a bit, the odd being used to set a bit. Therefore, locations FFC0 - FFC5 control three internal SAM bits. These addresses (C0 - C5) are used to control the vertical resolution. Writing to C3 and C5 will give a vertical resolution of 192 pixels (MODE 3 + 4).

The last area involved with graphics is location FFC8 - FD. These control the area used for graphics. Usually, the area used by graphics is &H600 - &H1Dff and, therefore, locations FFC7 (H200 offset) and FFC9 (&H400 offset) is set. If you wish to relocate graphics the SAM control bits of each give an offset of &H200 x bit number, i.e., each 3 will give an offset of &H600.

In brief, to obtain PMODE 4,1 Screen 1,0, the following locations have to be written to:-

PMODE 4

FF22 → F0
FFC3 → any
FFC5 → any
FFC7 → any
FFC9 → any

PMODE 3

FF22 → E0
FF22 → E0
FF22 → E0
FF22 → E0
FF22 → E0

M O D E 2 4

Dragon BASIC supports an extensive number of graphics modes. There are, however, a number of extra modes available to the machine code programmer. The most useful of these is Mode 24, which allows text and nine-colour hi-res graphics to be mixed on the screen.

To access this mode you will first need to:-

1. Set the VDG.

This can be done from BASIC by entering SCREEN 0,0.

2. Set the SAM

```
POKE &HFFC0,0
POKE &HFFC3,0
POKE &HFFc5,0
```

The MODE 24 screen is mapped from &H400 to &H1C00. Each location controls a block on the screen which is eight pixels (horizontally) by one pixel (vertically).

Each block is further divided in half, giving two 4 x 1 elements. These elements may not be controlled uniquely, but may be manipulated if consideration is given to the effect this will have on the other element in the block.

Each 8 x 1 block is a subset of a particular 8 x 12 matrix. These matrices correspond to those used to compose characters on the text screen. The effect of writing a byte to a location on the Mode 24 screen depends upon:-

- a. The position of the block within the character matrix.
- b. The character code that is equivalent to the byte written.

Every character with a code between 32 and 255 has a unique 12 x 8 matrix to determine how it is displayed on the screen. In general, if a Mode 24 location corresponds to the nth row of a text screen position, writing code X to that location results in the nth row of the character X matrix being displayed on the screen.

Thus it is possible to build text characters in Mode 24 by constructing them from their constituent blocks, or to create hi-res graphics from the same basic structures.

WWW.DRAGONDATA.CO.UK