

# Assembler für Fortgeschrittene#

von Uwe Röder, CSM 02/1990

In diesem Teil unseres Kurses werde ich mich mit der Anwendung von veränderten Display-Lists in Maschinen-Sprache beschäftigen.

Wir alle kennen wohl die großen Vorteile, die uns die Display-List bietet. So können wir mit ihrer Hilfe dem Bildschirm fast jedes beliebige Aussehen geben, indem wir verschiedene Grafikstufen mischen können.

Da in unserem Magazin schon sehr oft auf die einzelnen Befehle der Display-List eingegangen wurde, werde ich diese hier nur sehr kurz anführen:

Die Display-List Befehle bestehen alle aus einem Byte. Bei einem Sprung Befehl und dem LMS-Befehl (Bildschirmspeicher festlegen) kommen dazu noch zwei Bytes, die die benötigte Adresse angeben.

Die unteren vier Bits (0-3) eines dieser 'Befehls-Bytes' geben über die Grafikstufe Auskunft. Die oberen vier Bits (4-7) legen fest ob horizontal oder vertikal gescrollt werden soll, oder ob die Bildschirmspeicheradresse geladen werden soll, oder ob ein Display-List Interrupt ausgeführt werden soll.

Unteres Nibble (Bits 0 bis 3)

0 - Ausgabe von Leerzeilen auf dem Bildschirm. Das High-Nibble gibt in diesem Fall nur Auskunft über die Anzahl der Zeilen und den DLI. Ist Bit 7 gesetzt, so wird ein DLI ausgeführt. Die Bits 4 bis 6 geben Auskunft über die Anzahl der Zeilen:

Bits 4 bis 6:

000	-	1	Leerzeile
001	-	2	
010	-	3	
011	-	4	
100	-	5	
101	-	6	
110	-	7	
111	-	8	

1 Sprungbefehl

Ist Bit 7 gesetzt, so wird auch hier ein DLI ausgelöst. Ist Bit 6 gesetzt, so wird auf den nächsten VBI gewartet. Dieser Sprung sollte ganz am Ende einer Display-List stehen und zum Anfang der Display-List zurückspringen. Ist Bit 6 nicht gesetzt, so wird nur ein Sprung an einen anderen Speicherbereich durchgeführt, an welchem sich die Display-List fortsetzt. Dies muss teilweise erfolgen, da die Display-List keine 1 KB Grenze überschreiten darf.

Es folgen nun die verschiedenen Grafikstufen. Es wird dabei immer angegeben wie viele Bytes pro Zeile benötigt werden, wie viele Farben zur Verfügung stehen und wie viele Bildschirmzeilen für eine Grafikzeile benötigt werden.

2	-	Text	40,2,8
3	-	Text	40,2,10
4	-	Text	40,5,8
5	-	Text	40,5,16
6	-	Text	20,5,8

```

7 - Text      20,5,16
8 - BitMap   40,4,8
9 - BitMap   80,2,4
A - BitMap   80,4,4
B - BitMap  160,2,2
C - BitMap  160,2,1
D - BitMap  160,4,2
E - BitMap  160,4,1
F - BitMap  320,2,1

```

Die Bits des oberen Nibble (Bits 4 - 7) haben folgende Bedeutung:

```

Bit 4: gesetzt: Horizontales Scrolling möglich
Bit 5: gesetzt: Vertikales Scrolling möglich
Bit 6: gesetzt: Bildschirmspeicheradresse wird geladen
Bit 7: gesetzt: DLI wird ausgelöst

```

Soviel zu den einzelnen Befehlen. Es ist nicht zwingend vorgeschrieben die Bildschirmspeicheradresse ganz zu Anfang der DL zu laden. Dies kann auch ganz am Ende der DL geschehen. Wichtig ist nur, dass mindestens einmal im Verlaufe der DL die Bildschirmspeicheradresse geladen wird. Geschieht dies nicht, so führt das lediglich zu einem seltsamen Flackern irgendwelcher undefinierbarer Bildschirminhalte.

Um in MS eine eigene Display-List zu benutzen bietet sich der Pseudo-Opcode .HX an, mit dessen Hilfe ich Hex-Werte in dem Programm ablegen kann. Eine ganz normale Display-List sieht in MS dann so aus:

```

00010          .LI OFF
00020 -----
00030 S          LDA #0
00040          STA $D40E
00050 ; INTERRUPTS SPERREN
00060          LDA #DL
00070          STA $230
00080          LDA /DL
00090          STA $231
00100          LDA #$40
00110          STA $D40E
00120 ; INTERRUPTS FREIGEBEN
00130 -----
00140 DL          .HX F0F0F0
00150 ; 24 LEERZEILEN
00160          .HX 4240BC
00170 ; BILDSCHIRMSPEICHER NACH BC40
00180          .HX 0202020202020202
00190          .HX 0202020202020202
00200          .HX 0202020202020202
00210 ; INSGESAMT 24 * GR. 0
00220          .HX 41
00230          .DA DL
00240 ; SPRUNG AN DL-ANFANG
00250 -----

```

Das wäre eigentlich schon alles. Als MS-Programmierer muss man eigentlich nur darauf achten welchen Speicherbereich man belegt. Wenn man z.B. mehr als 24 Gr.0 Zeilen benutzt, so reicht der normale Speicherbereich ab BC20 oder 9C20 nicht mehr aus und man muss sowohl die DL als auch den Bildschirmspeicher woanders unterbringen. Zu beachten ist in diesem Fall unbedingt, dass eine DL keine 1KB Grenze überschreiten darf. Wichtig ist auch, dass vom Ende der DL an den

Anfang zurückgesprungen wird. Die DL ist nämlich im Grunde genommen eine Endlosschleife, die pro Bildschirmaufbau (50 mal pro Sekunde) abgearbeitet wird.

Falls die Informationen über die DL nicht ausreichen sollten kann, ich nur noch auf die vielen Berichte oder Lehrgänge verweisen, die bisher in unserem Magazin erschienen sind. Einige einfache Beispiele befinden sich auch im Anhang. Diese sind im Bibo-Assembler Format abgelegt.

---

## Anhang#

### Beispiel 1 (DL1.ASM)#

```
00010          .LI OFF
00020 -----
00030 S        LDA #DL
00040          STA $D40E
00050          LDA #DL
00060          STA $230
00070          LDA /DL
00080          STA $231
00090          LDA #$40
00100          STA $D40E
00110          RTS
00120 -----
00130 DL        .HX 304240BC
00140          .HX 0002000200020002
00150          .HX 0002000200020002
00160          .HX 0002000200020002
00170          .HX 0002000200020002
00180          .HX 0002000200020002
00190          .HX 0002000200020002
00200          .HX 0002
00210          .HX 41
00220          .DA DL
00230 -----
00240 ; 4 LEERZEILEN
00250 ; 24 LEERZ.+26*GR.0 IM WECHSEL
00260 -----
```

### Beispiel 2 (DL2.ASM)#

```
00010          .LI OFF
00020 -----
00030 S        LDA #DL
00040          STA $D40E
00050          LDA #DL
00060          STA $230
00070          LDA /DL
00080          STA $231
00090          LDA #$40
00100          STA $D40E
00110          RTS
00120 -----
00130 DL        .HX 304240BC
00140          .HX 02020202
00150          .HX 02020202
00160          .HX 02020202
00170          .HX 02020202
```

```

00180          .HX 02020202
00190          .HX 02020202
00200          .HX 02020202
00210          .HX 41
00220          .DA DL
00230 -----
00240 ; 4 LEERZEILEN + 29 * GR. 0
00250 -----

```

### Beispiel 3 (DL3.ASM)#

```

00010          .LI OFF
00020 -----
00030 S        LDA #DL
00040          STA $D40E
00050          LDA #DL
00060          STA $230
00070          LDA /DL
00080          STA $231
00090          LDA #$40
00100          STA $D40E
00110          RTS
00120 -----
00130 DL        .HX 304240BC
00140          .HX 02020202
00150          .HX 02020202
00160          .HX 02020202
00170          .HX 02020202
00180          .HX 02020202
00190          .HX 020202
00200          .HX 4240BC
00210          .HX 02020202
00220          .HX 41
00230          .DA DL
00240 -----

```

### Beispiel 4 (DL4.ASM)#

```

00010          .LI OFF
00020 -----
00030 S        LDA #DL
00040          STA $D40E
00050          LDA #DL
00060          STA $230
00070          LDA /DL
00080          STA $231
00090          LDA #$40
00100          STA $D40E
00110          RTS
00120 -----
00130 DL        .HX 30
00140          .HX 4240BC
00150          .HX 02020202020200
00160          .HX 4240BC
00170          .HX 02020202020200
00180          .HX 4240BC
00190          .HX 02020202020200
00200          .HX 4240BC
00210          .HX 02020202020200

```

```
00220      .HX 41
00230      .DA DL
00240 -----
```

### Beispiel 5 (DL5.ASM)#

```
00010      .LI OFF
00020 -----
00030 S      LDA #DL
00040      STA $D40E
00050      LDA #DL
00060      STA $230
00070      LDA /DL
00080      STA $231
00090      LDA #$40
00100      STA $D40E
00110      RTS
00120 -----
00130 DL      .HX F0F0F0
00140      .HX 4210BF
00150      .HX 02020202
00160      .HX 4220BE
00170      .HX 02020202
00180      .HX 4230BD
00190      .HX 02020202
00200      .HX 4240BC
00210      .HX 02020202
00220      .HX 41
00230      .DA DL
00240 -----
```