

Einführung in Z-80-Maschinensprache:

# Klartext für den ZX 81

## Teil 1: Daten lesen und verändern

Als Voraussetzung für das Programmieren in Maschinensprache bietet der ZX 81 zwei Basic-Befehle, die den Zugriff auf einzelne Bytes im Speicher erlauben.

Noch vor wenigen Jahren waren sogenannte Heimcomputer so beschaffen, daß man beim Programmieren nicht um Maschinensprache, also um die Befehlseingabe in hexadezimaler Form, herumgekommen ist. Moderne Heimcomputer verstehen hingegen Basic und können ohne jede Kenntnis von Maschinensprache gut programmiert werden. Aber die Maschinensprache lebt trotzdem weiter. Das zeigt sich bei einem Blick in Mikrocomputer-Zeitschriften oder beim Auflisten professionell geschriebener Programme und hat gute Gründe.

Als Einsteiger freilich steht man dem Maschinencode ratlos gegenüber, und die Bedeutung von Programmzeilen mit seltsamen REM-Kommentaren bleibt im Dunkeln. Wer sich aktiv mit seinem Heimcomputer beschäftigt und dessen Leistungsfähigkeit ausschöpfen möchte, wird aber gewiß nicht lange in

der „Schreckstarre“ verharren, sondern den Dingen auf den Grund gehen. Dabei will die Serie „Klartext für den ZX 81“ Schrittmacherdienste leisten und leicht verständlich mit vielen praktischen Übungen das Programmieren in Maschinensprache verständlich machen. Da besonders Einsteiger angesprochen sind, wurde der ZX 81 als Übungsmodell ausgewählt.

### Maschinenprogramme sorgen für Tempo

Maschinensprache ist eine elementare Sprache, auf die andere, höhere Programmiersprachen aufbauen. So wurde von den Entwicklern des ZX 81 erst einmal ein Maschinenprogramm geschrieben, das dem Computer die

Programmiersprache Basic verständlich macht. Es wurde im ROM (Lesespeicher) des ZX 81 verankert und kann nicht verändert werden. Wenn Sie also ein Basic-Programm eingegeben haben und es durch RUN zum Laufen bringen, wird jedes Zeichen, jedes Schlüsselwort vom ROM übersetzt, d. h. interpretiert: Man nennt das Maschinenprogramm im ROM deshalb „Interpreter“.

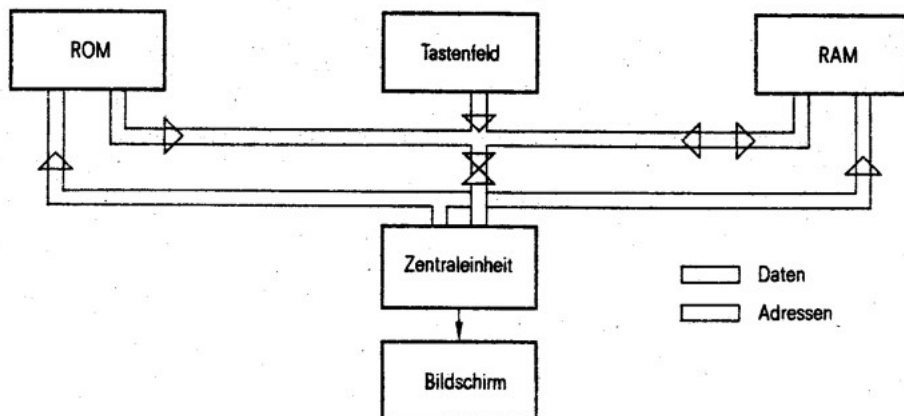
Der Interpreter macht der Zentraleinheit im Computer verständlich, was die Basic-Programmschritte bezwecken sollen. Leider nimmt dieses andauernde Interpretieren viel Zeit in Anspruch, was sich besonders bei schnellen Spielen, bei bewegten Grafiken, beim Rechnen und überall dort, wo Zeit kostbar ist, störend bemerkbar macht. Es muß also gelingen, den Interpreter zu übergehen und gleich ein für den Computer unmittelbar verständliches Maschinenprogramm anstelle des Basic-Programms zu schreiben. Doch bevor wir damit beginnen können, ist etwas Theorie notwendig.

### Die Hausnummern der RAM-Straße

Neben dem ROM hat der ZX 81 auch noch ein RAM (Arbeitsspeicher). Wie Bild 1 zeigt, lassen sich hier im Gegensatz zum ROM die Daten auch verändern. Selbstgeschriebene Maschinenprogramme können also nur dort abgelegt werden. Doch wie sieht nun so ein Speicher aus?

Ein Speicher besteht aus vielen einzelnen Zellen, in die jeweils ein Bit paßt. Damit man sie gezielt ansprechen kann und nicht verwechselt, werden jeweils 8 Bit zu einer Gruppe zusammengefaßt (1 Byte) und mit einer Art Hausnummer versehen, nämlich einer Adresse. Ein gespeichertes Programm ist in solchen 8-Bit-Speicherzellen untergebracht. Sehr wichtig ist, daß jedes Byte dann eine Dezimalzahl von 0 bis 255 darstellen kann ( $255 = 2^8 - 1$ ; Eins wird abgezogen, weil die Zählung bereits mit Null beginnt).

Betrachten wir jetzt den Aufbau des Arbeitsspeichers (RAM) näher: Sein Bereich beginnt, wie aus Bild 2 ersichtlich, bei Adresse 16 384. Ohne Arbeitsspeicher reicht er 1024 Byte (1 KByte)



① **Blockschaltung eines Computers:** Die Zentraleinheit kann mit 16-Bit-Adressen max. 65 535 8-Bit-Speicherzellen im RAM adressieren