

EPROM's, oder ROM's, sind eine sehr nützliche Sache. Eingesetzt werden sie immer dann, wenn ein Programm sofort nach dem Einschalten des Computers zur Verfügung stehen soll, oder wenn die Ladezeiten von Diskette zu langsam sind, oder wenn man Ladefehler durch den Benutzer ausschließen will.

In Ihrem ATARI Computer befinden sich gleich zwei ROM's. Das Betriebssystem und das BASIC. Natürlich wäre es auch möglich das Betriebssystem oder das BASIC von einer Diskette zu laden. Bei manchen Computertypen wird das auch gemacht. Zum Beispiel beim IBM PC, oder, am Anfang, auch beim ATARI ST. Aber diese Ladezeiten sind einfach zu lang.

Viel komfortabler ist es, wenn Sie Ihren Computer einschalten und dieser meldet sich sofort. Dafür sorgen die ROM's mit dem Betriebssystem und dem BASIC.

Was aber ist nun eigentlich ein ROM, oder ein EPROM? Grundsätzlich unterscheiden wir drei verschiedene Arten von Speicherbausteinen in unseren Computern.

1. Das RAM

RAM ist eine Abkürzung und bedeutet RANDOM ACCESS MEMORY (FREI ZUGREIFBARER SPEICHER). Oder verständlicher gesagt, Schreib-Lesespeicher. Und das sagt auch gleich, was wir mit dem RAM anstellen können. Das RAM ist der Arbeitsspeicher unseres Computers. Er dient zur Aufnahme von Daten, die sich während des Programmablaufes ändern, bzw. des Programmes überhaupt. Der Computer braucht das RAM zum aufbauen von Displaylisten, zum ablegen von Grafikdaten, für Timer, Tabellen, Stapelregister, Programme usw. Die Aufgaben des RAM-Speichers sind so groß, daß wir in einem gesonderten Artikel noch einmal ausführlich darauf eingehen werden.

2. Das ROM

ROM ist ebenfalls eine Abkürzung und steht für READ ONLY MEMORY (NUR LESE SPEICHER). Damit ist auch gleich gesagt, wozu ROM's benötigt werden. Sie enthalten ausschließlich Programme und nicht veränderbare Daten. Beim Einschalten des Computers stehen diese Programmdateien dann sofort zur Verfügung. Der Computer kann dann mit den Programmdateien arbeiten. ROM's können nur einmal programmiert werden. Sie können nicht wieder gelöscht werden.

3. Die EPROM's

Auch EPROM ist eine Abkürzung. Es steht für ERASABLE PROGRAMMABLE READ ONLY MEMORY (LÖSCHBARER, PROGRAMMIERBARER NUR LESE SPEICHER). Damit wird dann auch schon der Unterschied zwischen einem ROM und einem EPROM deutlich. Im Gegensatz zum ROM kann das einmal programmierte EPROM wieder gelöscht und neu programmiert werden.

Wenn Sie jetzt einmal in Ihren ATARI Computer hinein sehen, werden Sie feststellen, daß ATARI zwei ROM's für das Betriebssystem und das BASIC benutzt hat. ROM Bausteine werden von Herstellern den EPROM's vorgezogen, weil sie in größeren Stückzahlen billiger herzustellen sind als EPROM's. Das Programm, oder die Daten, werden bei der Herstellung des ROM's bereits als Maske fest einprogrammiert. Der zweite Vorteil eines ROM's ist, daß es nicht durch UV-Licht gelöscht werden kann. Gerade deswegen interessieren uns aber die EPROM's, da es die einzigen Bausteine sind, die jeder Programmierer selber programmieren kann (wenn er die entsprechende Hardware hierzu besitzt).

Um die nachfolgenden Daten richtig zu verstehen, sehen wir uns zuerst einmal die folgende Grafik an.

Die Grafik zeigt die Pinbelegung eines EPROM's in einem 28 poligen Gehäuse. Anwendbar ist die Grafik für die EPROM Typen 2764 (8K), 27128 (16K) und 27256 (32K).

Einziger Unterschied: Anschlußbeinchen 26 (16K) und 27 (32K). Diese werden bei dem Typ 2764 auf +5 Volt gelegt. Beim Typ 27128 wird nur der Pin 27 auf +5 Volt gelegt.

Die Abkürzungen bedeuten:

VPP = Programmierspannung
A0-A13 = Adressleitungen 0-13
D0-D7 = Datenleitungen 0-7
GND = Masse 0 Volt
OE = Output Enable
CE = Chip Enable
VCC = Betriebsspannung +5 Volt

EPROM's sind immer zu 8-Bit Datenbreite organisiert, der Unterschied besteht nur in der Anzahl der Speicherstellen. Ausgestorben sind inzwischen die EPROM Typen 2708/2758 mit einer Kapazität von 1K-Byte. Noch gängige Typen sind 2516/2716 (2K-Byte) und 2532/2732 (4K-Byte), die aber verhältnismäßig teuer sind und selten verwendet werden. Diese sind die einzigen EPROM's im 24 poligen Gehäuse und man braucht 25 Volt um sie zu programmieren. Die 25er Typen unterscheiden sich von den 27er Typen nur durch die Anschlußbelegung.

Die gängigsten EPROM's sind die Typen 2764 (8K-Byte), 27128 (16K-Byte) und 27256 (32K-Byte). Alle diese EPROM Typen haben ein 28 poliges Gehäuse. Die Programmierspannung kann heute leider nicht mehr allgemein angegeben werden. Hier gilt die Regel: Befindet sich kein Spannungsaufdruck auf dem Gehäuse des EPROM's, gilt bei den Typen 2764 und 27128 die Spannung von 21 Volt, bei den Typen 27256 die Programmierspannung 12,5 Volt.

Neu sind die Stromsparenden C-MOS EPROM's der Typen 27C16, 27C32, 27C64, 27C128 und 27C256. Sie sind den "normalen" EPROM's vorzuziehen, da sie eine geringere Stromaufnahme haben und das Netzteil des Computers nicht so stark belasten.

WIE WIRD EIN EPROM PROGRAMMIERT?

Wir werden den programmiervorgang an Hand eines EPROM's des Types 2764 erklären.

Zum programmieren eines EPROM's brauchen Sie einen sogenannten EPROM-BURNER (EPROM-BRENNER). Normalerweise bekommen Sie von den hier beschriebenen Arbeiten nichts mit, da die Steuersoftware des Eprommers alle Arbeiten automatisch ausführt.

Alle EPROM's werden vom Hersteller in der Regel gelöscht geliefert. Das bedeutet, alle Bits sind auf 1 gesetzt. Somit sind alle Bytes auf \$FF (255) gesetzt. Lesen Sie dieses EPROM mit dem Burner aus, bekommen Sie nur \$FF als Inhalt aller Speicherstellen. Sollte ein EPROM einmal nicht gelöscht geliefert werden, müssen Sie diese Arbeit selber ausführen.

Nun wird das EPROM in den Sockel des Burners eingelegt und Sie laden die Daten, die Sie in das EPROM brennen möchten. Nachdem Sie den Befehl zum programmieren des EPROM's gegeben haben, beginnt für die Steuersoftware des Burners die Arbeit.

Für jedes Bit enthält ein EPROM eine Feldeffekt Transistor Zelle. Der Gate Anschluß dieser Zelle "hängt in der Luft" (FLOATING). Im unprogrammierten Zustand wird dadurch eine logische 1 dargestellt. Bringt man nun eine Spannung an diese Bitzelle, schaltet der Transistor durch und die logische 1 wird in eine logische 0 umgewandelt. Um nun ein Programm in ein EPROM zu brennen, müssen alle entsprechenden Bits der benötigten Bytes auf logisch 0 gesetzt werden. Dieses wird durch die folgenden Arbeitsgänge erreicht:

1. Im EPROM wird ein Wort adressiert.
2. Die Chip Auswahlleitung wird auf logisch 0 gesetzt und die Programmierspannung wird durchgeschaltet.
3. Die Daten werden parallel auf die Datenleitungen des EPROM's gelegt.
4. Nach Stabilisierung der Daten und Adressensignale wird ein 50 Millisekunden langer langer TTL-Impuls zum Programmieranschluß gesendet. Alle Bits, die logisch 1 behalten sollen, bleiben unverändert. Alle anderen werden dabei auf logisch 0 gesetzt.

Bei den neuen Programmiergeräten ist meist ein Fast-Programmiermodus fest eingebaut. Das bedeutet, statt mit einer festen Programmierdauer von 50ms wird hier mit Programmierimpulsen von 1ms gearbeitet. Dabei werden die Daten nach jedem Impuls mit den Originaldaten verglichen. Stimmen die Daten überein, wird noch einmal mit der dreifachen bisher benötigten Zeit nachprogrammiert. Dieses Verfahren bringt eine sehr große Zeiteinsparung. Der Fast-Programmiermodus kann angewendet werden, da alle EPROM's in der Regel bereits nach wenigen Impulsen programmiert sind.

Alle Schritte müssen für jede einzelne Speicherstelle (Byte) des EPROM's einzeln durchgeführt werden. Bei der Programmierung von Blöcken oder ganzen EPROM's wird die Adresse durch einen Adresszähler automatisch erhöht.

WIE LÖSCHT MAN EIN EPROM?

Der Vorteil eines EPROM's gegenüber eines ROM's liegt in der Wiederverwendbarkeit. Zu diesem Zweck können EPROM's gelöscht werden. Dies geschieht durch intensive Bestrahlung durch UV-Strahlen. Jedes EPROM besitzt auf der Oberseite ein rundes Quartzfenster. Nach erfolgreicher Programmierung sollte dieses Fenster möglichst abgeklebt werden. Denn bereits die geringe UV-Strahlung des Sonnenlichtes kann ausreichen, ein EPROM zu löschen. Aber keine Bange, ein EPROM das dem normalen Tageslicht (Sommer, Sonne) ausgesetzt ist, verliert erst nach etwa einem Monat die Daten.

Löschen bedeutet im übrigen nichts anderes, als alle Bits wieder auf logisch 1 zu setzen. Zum löschen unserer EPROM's verwenden wir einen Ozonfreien Strahler mit einer Wellenlänge von 230nm. Übrigends, Heimsonnen eignen sich nicht zum löschen von EPROM's!

Die Zeit die wir benötigen, um ein 2764er EPROM zu löschen, können wir mit der folgenden Formel ausrechnen:

$$t = \frac{1500 \frac{\text{mWs}}{\text{cm}}}{12 \frac{\text{mW}}{\text{cm}}} * \frac{1}{60} = 20,8 \text{ Min}$$

Die Daten für andere EPROM Typen entnehmen Sie bitte den Datenblättern, die jeder EPROM Hersteller für seine Produkte heraus gibt. Aber Grundsätzlich gilt: Je stärker die Leistung der Löschlampe, desto kürzer die Löschzeit. Gewöhnliche im Handel erhältliche Löschlampen benötigen nur etwa 5min um ein EPROM vollständig zu löschen.

EINIGE TIPS ZUM UMGANG MIT EPROMS

Das EPROM muß unbedingt richtig in der Fassung des Burners sitzen.

Unbenutzte EPROM's verwahrt man entweder in einer antistatischen Röhre oder steckt sie auf antistatischen Schaumstoff auf.

Kleben Sie unbedingt das Fenster nach dem programmieren zu!

Achten Sie immer darauf, daß Sie die EPROM's richtig herum einsetzen! Sowohl auf dem Burner, als auch in den Sockeln!

WELCHE PROGRAMME SIND EPROM GEEIGNET?

Leider können nicht alle Programme in ein EPROM gebrannt werden. Grundsätzlich gilt folgendes:

1. Das Programm muß in Maschinensprache geschrieben sein.
2. Das Programm muß speziell für den Betrieb mit einem EPROM geschrieben worden sein. Das heißt, der Programmierer muß sein Programm für den Betrieb in einem EPROM vorgelesen haben. (Viele wollen das TURBO BASIC XL in ein EPROM brennen und es gegen das ATARI BASIC austauschen. DAS GEHT NICHT!)

Die ATARI Computer stellen uns einen maximal 16K-Byte großen Speicherplatz für ein ROM Modul zur Verfügung. Dieser Speicherplatz liegt zwischen den Adressen \$8000 und \$BFFF (16K) oder \$A000 und \$BFFF (8K). Der Speicherbereich ist folgendermassen aufgebaut:

| | 8K-ROM | 16K-ROM |
|-----------------------|--------|---------|
| ROM INIT-ADRESSE HIGH | \$BFFF | \$BFFF |
| ROM INIT-ADRESSE LOW | \$BFFE | \$BFFE |
| OPTIONS-BYTE | \$BFFD | \$BFFD |
| KENN-BYTE | \$BFFC | \$BFFC |
| ROM-STARTADRESSE HIGH | \$BFFB | \$BFFB |
| ROM-STARTADRESSE LOW | \$BFFA | \$BFFA |
| ANWENDERPROGRAMM | | |
| ROM ANFANG | \$A000 | \$8000 |

An Hand der Tabelle sehen Sie, daß das Betriebssystem außer dem Programm noch einige Angaben über das Programm benötigt. Es muß wissen:

1. Soll beim Einschalten des Computers gebootet werden?
2. Wo liegt die Startadresse des Programmes?
3. Soll das Modul nur initialisiert, oder soll es auch gestartet werden?
4. Gibt es eine Initialisierungsadresse?

Alle diese Informationen erhält das Betriebssystem aus den letzten sechs Bytes des ROM Modules.

Zum Beispiel die Adresse \$BFFC. Steht nach dem Einschalten des Computers in der Speicherstelle \$BFFC ein anderer Wert als \$00, leitet es daraus ab, das sich in diesem Bereich (\$A000 bis \$BFFF) weder RAM noch ROM befindet.

Das OPTIONS-BYTE in \$BFFD gibt dreierlei Informationen:

| | | | | | | | | |
|---------|---|---|---|---|---|---|---|---|
| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| benutzt | x | - | - | - | - | x | - | x |

Bit 7 = 1 Sagt dem Betriebssystem, daß es sich bei dem ROM um ein Diagnosemodul handelt. Vor der Betriebssystem-Initialisierung wird das Modul initialisiert (JMP (\$BFFE)).
= 0 Dies ist ein normales ROM-Modul.

Bit 2 = 1 Initialisiere und Starte das ROM-Modul.
= 0 Initialisiere das ROM-Modul, starte es aber nicht.

Bit 0 = 1 Boote die Diskette.
= 0 Nicht booten.

Beispiel:

\$BFFD = \$03 = %00000100 bedeutet: Dies ist ein "normales" ROM-Modul, Initialisiere und starte es, Diskette wird nicht gebootet. Diesen Wert finden wir zum Beispiel bei Spielmodulen.

Wohin der Computer nach drücken der SYSTEM RESET Taste oder nach dem Einschalten mit einem JSR springen soll, erfährt das Betriebssystem aus der ROM-ADRESSE \$BFFE/\$BFFF. Die Adresse ist im LOW/HIGH Format abgelegt. In der Regel zeigt diese Adresse auf einen RTS Befehl.

Für das Betriebssystem wichtiger ist die ROM-STARTADRESSE \$BFFA/\$BFFB. Das ist die Adresse, die nach dem Einschalten des Computers angesprungen wird, sofern im OPTIONS-BYTE \$BFFD Bit 2 auf 1 gesetzt ist.

Mit diesen Angaben sollten Sie jetzt eigentlich in der Lage sein, ein eigenes Programm für ein EPROM zu schreiben. Natürlich wird es noch einige Probleme geben, wenn Sie zum Beispiel ein Programm in ein EPROM brennen wollen, das mehr als 16K-Byte Speicherplatz beansprucht. Das, und noch einige Tips und Tricks, soll Thema eines später erscheinenden Artikels werden. Bis dahin wünschen wir Ihnen viel Spaß beim programmieren!