

**Name:**  
\_\_\_\_\_**GAL-Programmiersoftware****Datum:** \_\_\_\_\_

---

## 5.0 Einsatz der Programmiersoftware

### 5.1 Hardwarevoraussetzungen

Die Software liegt in Form einer Diskette (3,5 oder 5,25") vor und muß ab Adresse C000 H betrieben werden.

Zur Anwendung der Software benötigt man 4-K-Byte RAM ab Adresse D000 H. Günstig ist hierbei die Verwendung der 64-K-Speicherkarte mit dem MAT 32K.

### 5.2 Programmaufruf

Die GAL-Programmiersoftware wird mittels des GO-Kommandos gestartet. Hierzu rufen Sie das GO-Kommando auf und geben die Startadresse der Programmiersoftware, C000 H, ein.

Hierdurch wird folgende Meldung auf dem Bildschirm erzeugt:

GAL 16V8, 16V8A V2.3

EDIT  
PROG  
DELETE  
HELP  
TEST  
READ  
LPRINT  
VERIFY  
CHANGE  
NEW  
QUIT

KMD\*>

Dieses ist die Liste der unter dem GAL-Programmierer möglichen Kommandos. Das Prompt ist an dem zusätzlichen Zeichen "\*" von den Kommandoaufrufen KMD> (MAT 85) und KMD+ (MAT 85+) zu unterscheiden.



**Name:**

---

**GAL-Programmierinterface****Datum:**

---

Die Kennzeichnung "X" bedeutet hierbei, daß eine Verbindung programmiert werden soll. Ein "-" bedeutet, daß hier später keine Verbindung programmiert wird. Die Software schlägt also die Programmierung aller möglichen Verknüpfungen vor. Der Vorschlag für jede Verbindung kann durch Überschreiben mit dem Zeichen "-" geändert werden und auch, durch erneute Eingabe von "X" wieder zurückgeändert werden. Hierzu muß die Schreibmarke, das ist der blinkende Unterstrich, mit den Pfeiltasten auf die Stelle in der Tabelle bewegt werden, die geändert werden soll.

**Steuerfunktionen innerhalb des Editiermodus:**

Betätigung der Taste(n) :

- |                        |  |
|------------------------|--|
| <b>Pfeiltasten</b>     | bewegen die Schreibmarke   |
| <b>Buchstabe "P"</b>   | führt zu den Produkterm-Verbindungen (PTD = Produkterm disable). Ein "X" an dieser Stelle sperrt die UND-Verknüpfung, unabhängig davon, wie deren übrige Eingänge programmiert sind. |
| <b>Leerzeichen</b>     | führt zum nächsten AND-Array, bzw. nach AND-Array 7 zum Ausgangs-Steuerwort.   |
| <b>Carriage Return</b> | führt zurück zur KMD*>-Kommandoebene   |

**6.1.2 Editieren des Ausgangs-Steuerwortes**

Mit dem Ausgangs-Steuerwort wird die Makrozelle des GAL-Bausteins beeinflusst. Die Bedeutung der einzelnen Bits in dem Steuerwort wird hier nur kurz aufgeführt.

Name: \_\_\_\_\_

GAL-Programmierinterface

Datum: \_\_\_\_\_

Um das Steuerwort zu editieren geben Sie , nach dem Aufruf des Editierkommandos, den Buchstaben A gefolgt von (CR) ein.

KMD\*> EDIT? A(CR) Hierauf erscheint die folgende Meldung auf dem Bildschirm :

AUSGANGS-STEUERWORT

```
SYN ACO 0 AC1 7 0 XOR 7
0 0 0000 0000 0000 0000
```

Im Rechnervorschlag sind alle Bits des Ausgangs-Steuerwortes auf 0 eingestellt. Änderungen können durch das Überschreiben mit dem Wert 1, Rückänderungen durch Überschreiben mit 0 durchgeführt werden. Die einzelnen Bits können mit den Pfeiltasten für rechts, bzw. links ausgewählt werden.

#### Kurzbeschreibung der Bedeutung der Bits :

SYN = Synchronbit. 0 bedeutet, der Baustein arbeitet mit einem Takt, 1 bedeutet, der Baustein arbeitet rein kombinatorisch.

ACO und AC1 = Architecture Control Bit. Die Kombination dieser Bits beeinflusst die Makrozelle des GAL-Bausteins so, daß z.B. Ausgänge zu Eingängen geschaltet werden, bzw. festgelegt wird, womit die Ausgangsfreigabe (Output Enable) erfolgen soll, woher ggf. Rückführungen kommen sollen. ACO existiert nur einmal für den gesamten Baustein, während AC1 für jede der 8 Ausgangsmakrozellen programmierbar ist.

Name: \_\_\_\_\_

GAL-Programmierinterface

Datum: \_\_\_\_\_

---

XOR = Exclusive OR. Hiermit können die Ausgangszustände der ODER-Verknüpfungen invertiert werden. Eine 1 an der entsprechenden Position invertiert das Signal, eine 0 invertiert das Signal nicht.

**Steuerfunktionen innerhalb des Editiermodus:**

**Pfeiltasten** Schreibmarke nach rechts bzw. links

**Leerzeichen** führt zur Bausteinauswahl

**Carriage Return (CR)** führt zurück zur KMD\*>-Kommandoebene

**6.1.3 Editieren der Bausteinbezeichnung**

Als Bausteinbezeichnung kann ein Name oder eine Bezeichnung der Bausteinfunktion sowie eine Nummer dienen. Insgesamt darf eine solche Bezeichnung maximal 8 Zeichen lang sein.

Um eine Bausteinbezeichnung zu editieren geben Sie, nach dem Aufruf des Editierkommandos, den Buchstaben B gefolgt von (CR) ein.

KMD\*> EDIT? B(CR) Hierauf erhalten Sie die Meldung :

BAUSTEIN-BEZEICHNUNG

.....

Jeder dieser 8 Punkte kann durch einen Buchstaben, eine Ziffer oder ein Leerzeichen überschrieben werden. Die Bausteinbezeichnung wird beim Programmiervorgang des GAL-Bausteins mit programmiert.

**Steuerfunktionen innerhalb des Editiermodus:**

**Pfeiltasten** Schreibmarke nach rechts bzw. links

**Carriage Return (CR)** führt zurück zur KMD\*>-Kommandoebene



**Name:**  
\_\_\_\_\_**GAL-Programmierinterface****Datum:** \_\_\_\_\_

---

## 6.2 Das Programmierkommando PROG

Mit diesem Kommando wird der Programmiervorgang ausgelöst. Der durch das EDIT-Kommando in den RAM-Bereich für die "Fuse-Map" eingetragene Inhalt für die UND-Verknüpfungsfelder, das Ausgangs-Steuerwort, die Bausteinbezeichnung und das Securitybit wird in den GAL-Baustein programmiert.

Ein GAL-Baustein des Typs 16 V 8 oder 16 V 8 A muß hierzu in die Fassung des GAL-Programmierinterface gesteckt sein.

Aufruf des Kommandos :

KMD\*> P(CR)

Es erfolgt die Meldung :

KMD\*> PROG

CHANGE, PROG, VERIFY : bei fehlerfreier Programmierung.

Bei fehlerhaftem Programmierversuch werden Fehlermeldungen erzeugt.

## 6.3 Das Kommando DELETE

Dieses Kommando dient zum Löschen des Inhaltes eines programmierten GAL-Bausteins. Es werden alle programmierten Verbindungen, sowie Bausteinbezeichnung und Leseschutz zurückgesetzt.

Aufruf des Kommandos :

KMD\*> D(CR)

Es erfolgt die Meldung :

KMD\*> DELETE

Bei fehlerhaftem Versuch werden Fehlermeldungen erzeugt.

Name:

---

GAL-Programmierinterface

Datum:

---

#### 6.4 Das HELP-Kommando

Mit Aufruf dieses Kommandos wird die Kommandoliste nochmals ausgedruckt.

Aufruf des Kommandos

KMD\*> H(CR) Es erfolgt die Meldung :

Edit

.

Quit

KMD\*>

#### 6.5 Das TEST-Kommando

Das Test-Kommando dient dazu, daß die Baustein-Bezeichnung aus dem GAL ausgelesen und auf dem Bildschirm dargestellt wird.

Aufruf des Kommandos

KMD\*> T(CR) Es erfolgt die Meldung :

KMD\*> TEST

BAUSTEIN-BEZEICHNUNG

AMPEL 03

KMD\*> \_

Hierin ist die Bezeichnung AMPEL 03 ein Beispiel. Verfügt der Baustein über keine Bezeichnung, so wird diese Zeile als Leerzeile dargestellt.



Name: \_\_\_\_\_

GAL-Programmierinterface

Datum: \_\_\_\_\_

## 6.6 Das READ-Kommando

Mit diesem Kommando wird der Inhalt aus einem im GAL-Interface eingesteckten Baustein als Fuse-Map ins RAM eingelesen. Der eingelesene Inhalt kann jetzt z.B. bearbeitet oder auch in einen weiteren Baustein programmiert werden.

Das READ-Kommando kann nicht verwendet werden, wenn im Baustein das SECURITYBIT gesetzt ist. In diesem Fall können die Daten aus dem GAL-Baustein nicht gelesen werden !

Aufruf des Kommandos

KMD\*> R(CR) Es erfolgt die Meldung :

KMD\*> READ

KMD\*> \_

## 6.7 Das Kommando LPRINT

Mit diesem Kommando können alle Programmierdaten ausgedruckt werden.

Aufruf des Kommandos

KMD\*> L(CR) Es erfolgt die Meldung :

KMD\*> LPRINT? \_ An dieser Stelle kann, genau wie beim Editieren, ein Teil des AND-Array (0...7), die Baustein-Bezeichnung bzw. das Ausgangsteuerwort ausgewählt werden. Der jeweils ausgewählte Bereich wird auf Drucker oder Bildschirm ausgedruckt. Gibt man nichts weiteres an, sondern betätigt lediglich (CR), so wird der komplette Inhalt des RAM-Bereichs ausgedruckt also die Bausteinbezeichnung, alle acht AND-Arrays und das Ausgangs-Steuerwort.

Name: \_\_\_\_\_

GAL-Programmierinterface

Datum: \_\_\_\_\_

---

**6.8 Das VERIFY-Kommando**

Dieses Kommando dient zum Vergleich des RAM-Bereichinhaltes mit dem Inhalt eines im GAL-Programmierinterface eingesteckten GAL-Bausteins. Unterschiede werden hierbei gekennzeichnet.

Das VERIFY-Kommando kann nicht verwendet werden, wenn im Baustein das SECURITYBIT gesetzt ist. In diesem Fall können die Daten aus dem GAL-Baustein nicht gelesen werden und damit auch nicht verglichen werden !

Aufruf des Kommandos

KMD\*> V(CR) Es erfolgt die Meldung :

KMD\*&gt; VERIFY bei fehlerfreiem Vergleich, d.h. es besteht an keiner Stelle ein Unterschied zwischen RAM- und Bausteininhalt.

**6.8.1 Die Fehlermeldungen beim VERIFY-Kommando**

Werden durch das Verify-Kommando Unterschiede erkannt, so werden sie wie folgt gemeldet :

**6.8.1.1 Unterschied in der Bausteinbezeichnung**

Nach Aufruf des VERIFY-Kommandos erfolgt die Meldung :

KMD\*&gt; VERIFY

BAUSTEIN-BEZEICHNUNG

AMPEL 03  
IC ....RE 2RAM-Inhalt  
IC-Inhalt

Die Meldung zeigt erstens, daß in der Baustein-Bezeichnung Unterschiede gefunden wurden und zweitens, worin diese Unterschiede bestehen. Die Zeile, die mit "IC" beginnt, symbolisiert den Inhalt des GAL-Bausteins. In dem Beispiel sind bei den ersten vier Buchstaben der Baustein-Bezeichnung keine Unterschiede (...), aber bei den

Name: \_\_\_\_\_

GAL-Programmierinterface

Datum: \_\_\_\_\_

letzten vier sind im Baustein die Zeichen RE 2 statt L 03 (RAM) gefunden worden. Übereinstimmungen werden also durch Punkte und Unterschiede durch die entsprechenden Zeichen gekennzeichnet.

### 6.8.1.2 Unterschied im Ausgangs-Steuerwort

Unterschiede im Ausgangs-Steuerwort werden genauso wie die in der Bausteinbezeichnung dargestellt. Eine solche Fehlermeldung kann z.B. wie folgt aussehen :

KMD\*> VERIFY

#### AUSGANGS-STEUERWORT

SYN	AC0	0	AC1	7	0	XOR	7	
	1	0	0000	0000	1111	1111		RAM-Inhalt
IC	.	.	....	....	0...	....		IC-Inhalt

### 6.8.1.3 Unterschied im AND-Array

Nach Aufruf des VERIFY-Kommandos erscheint zunächst die Fehlermeldung :

KMD\*> VERIFY

\*\*\* AND-ARRAY UNGLEICH \*\*\*

KMD\*> \_

Jetzt gilt es, die Unterschiede im AND-Array ausfindig zu machen. Hierzu muß das EDIT-Kommando aufgerufen werden und die einzelnen Arrays müssen nun auf Fehlermeldungen durchsucht werden. Die Unterschiede werden in den UND-Verknüpfungsfeldern durch die Kennzeichnungen "0" und "1" angezeigt.

**Name:**

---

**GAL-Programmierinterface****Datum:**

---

Eine "0" besagt hierbei, daß im RAM eine Verbindung (X) eingetragen ist und im Baustein stattdessen keine Verbindung (-) programmiert ist.

Eine "1" signalisiert, im RAM ist keine Verbindung (-) eingetragen und im Baustein ist eine Verbindung (X) programmiert.

An den Positionen, an denen kein Unterschied besteht, sind die herkömmlichen Verbindungszeichen "-" und "X" eingetragen.

#### 6.9 Das Kommando CHANGE

Mit diesem Kommando werden die Fehlerkennzeichnungen, die zuvor erzeugt wurden wieder gegen die ursprünglichen RAM-Programmierdaten ausgetauscht. Das Kommando wird bei einem Programmiervorgang automatisch aufgerufen, um evtl. noch vorhandene Fehlermeldungen nicht irrtümlich mit zu übertragen. Ebenfalls werden die alten Fehlermeldungen beim erneuten Aufruf des Kommandos VERIFY durch das CHANGE-Kommando rückgängig gemacht.

Aufruf des Kommandos

KMD\*> C(CR)            Es erfolgt die Meldung :

KMD\*> CHANGE

KMD\*> \_

#### 6.10 Das Kommando NEW

Dieses Kommando dient dazu, die Programmierdaten in der Fuse-Map im RAM zurückzusetzen.

Name: \_\_\_\_\_

GAL-Programmierinterface

Datum: \_\_\_\_\_

---

Aufruf des Kommandos

KMD\*> N(CR)            Es erfolgt die Meldung :

GAL 16V8, 16V8A V2.3

KMD\*> \_

### 6.11 Das Kommando QUIT

Durch Aufruf dieses Kommandos wird der GAL-Programmierer verlassen und zur Kommandoebene des Betriebssystems zurückgekehrt.

Aufruf des Kommandos

KMD\*> Q(CR)            Es erfolgt die Meldung :

KMD\*> QUIT

KMD > \_

<p><b><u>Hinweis</u> :</b> Die Programmierdaten können mittels des SAVE und des LOAD-Kommandos des Betriebssystems gespeichert und zurückgelesen werden. Hierbei gilt als Startadresse D000 und als Stopadresse D89F</p>
--