

---

# **POSITIONIER- UND BAHNSTEUERUNG MCU-G3**

## **G3-ElCam Interface**



---

<b>1</b>	<b>Einführung</b>	<b>5</b>
<b>2</b>	<b>Verwendung der ELCAM-Funktionalität</b>	<b>6</b>
2.1	Initialisierung von ELCAM	6
2.2	Funktionen des ELCAM-Moduls	7
2.2.1	Ergänzende Infos zum Aktivierung des AxisCompensationMode	8
2.3	Multi-Line Tabellen	11
2.4	Fehlerabfrage von ELCAM	11
2.5	Weitere Eigenschaften der ELCAM Funktionalität	12
2.5.1	Tabellenwiederholung	12
2.5.2	Endgeschwindigkeit	12
2.5.3	Programmierung mehrerer Nachföhrtabellen	12
2.6	Verwendung von ELCAM	13
2.7	Übertragung kompletter Tabellen per PCI-Direktzugriff	13
2.7.1	Aufbau der EICAM-Tabelle mit 64bit Gleitpunktdarstellung	14
2.7.2	Aufbau der EICAM-Tabelle mit 32bit Gleitpunktdarstellung (Single Precision)	15
2.8	Ausführung von ELCAM-Tabellen im PC-Arbeitsspeicher	15
2.8.1	PCAP-Funktion allocPhysMem	15
2.8.2	PCAP-Funktion freePhysMem	16
2.8.3	Hinweise zur Verwendung von Physischem Speicher	16
2.8.4	Verwendung der ELCAM-Tabelleninterpolation mit Physischem Speicher	16
<b>3</b>	<b>Spindelsteigungs- und Winkelfehlerkompensation</b>	<b>17</b>
3.1	Spindelsteigungsfehlerkompensation	17
3.2	Winkelfehlerkompensation	17



# 1 Einführung

Die ELCAM-Funktionalität der MCU-G3 Produkte ermöglicht es einzelne Achsen des Systems über eine Tabellenfunktion einer Führungsachse nachzuführen. Hierzu wird eine Stützpunkt-Tabelle mit nahezu beliebiger Größe definiert. Zwischenwerte der Stützpunkte werden durch Linearinterpolation berechnet.

Die ELCAM-Funktionalität eignet sich für unterschiedliche Anwendungsbereiche:

- Multi-Line-Tabellen: Optional ist es möglich eine weitere Achse zu definieren, welche einen Zeilenindex für die Tabelle vorgibt. Somit ist es möglich eine zweidimensionale Tabelle zu definieren, über welche die Nachführachse, abhängig von zwei Führungsachsen gesteuert wird.
- Mehrere Tabellen: Es ist möglich, mehr als eine Tabelle zu definieren. Somit ist es möglich unterschiedliche Achsverbünde gleichzeitig zu aktivieren.
- Spindelsteigungsfehler-Kompensation: Mit Hilfe einer speziell konfigurierten Tabelle kann der Spindelsteigungsfehler einer Achse kompensiert werden.
- Winkelfehler-Kompensation: Mit Hilfe einer speziell konfigurierten Tabelle kann der Winkelfehler eines Achssystems kompensiert werden.

Die ELCAM-Funktionalität wird über das Universelle Object-Interface der MCU-G3 konfiguriert. ELCAM ist eine Option der MCU-G3 Betriebssystemsoftware RWMOS.ELF. Mit dem Dienstprogramm fwsetup kann, bei gebootetem System geprüft werden, ob das vorliegende RWMOS.ELF die Option unterstützt.

Defaultmäßig steht für die ELCAM-Tabellen ein Speicherbereich von 100.000 Bytes zur Verfügung. Falls dieser nicht ausreicht kann er mit der Umgebungsvariablen SZELCAMPUFFER erhöht werden. Der Maximalwert für diesen Speicher richtet sich nach der Speicher-Ausstattung der eingesetzten Steuerung. Bei sehr großen Tabellen (mehrere hundert Kilobyte bis einige Megabyte) kann die Ladedauer einige Minuten betragen. Um dies zu verhindern ist es möglich die komplette Tabelle per PCI-Memory Zugriff direkt vom PC-Arbeitsspeicher in den RAM-Arbeitsspeicher der Steuerung zu kopieren. Damit kann die Ladezeit erheblich verkürzt werden. Hierzu existieren gesonderte Ladebefehle.

## 2 Verwendung der ELCAM-Funktionalität

### 2.1 Initialisierung von ELCAM

Folgende Werte für das Universelle Object-Interface sind für die Verwendung des ELCAM-Moduls zu verwenden:

Tabelle 1: Object-Descriptor-Elemente

<b>Object-Descriptor Element</b>	<b>Wert</b>
<b>BusNumber</b>	1200
<b>DeviceNumber</b>	1 Bei Funktion 30 (siehe Tabelle 2) existiert auch eine Funktion für DeviceNumber = 0
<b>Index</b>	0, 1, ... Laufende Nummer der Tabelle (wird vom Benutzer vergeben). Jede Tabelle wird durch eine Nummer referenziert und kann unterschiedliche Achsverbände bedienen.
<b>SubIndex</b>	Funktionsnummer lt. Tabelle 2

Weitere Informationen zu den Object-Descriptor-Elementen sind im Dokument G3-Universelles-Objekt-Interface.pdf beschrieben.

## 2.2 Funktionen des ELCAM-Moduls

Tabelle 2: Funktionen ELCAM-Modul SubIndex (bzw. Index bei DeviceNr. 1 Befehl 30)

Nr.	Bez.	Typ	Erläuterung
1	ERROR	integer r/w	Fehlerstatus lesen / rücksetzen Bitcodierung siehe Tabelle 3
2	RESET	integer r/w	Tabelle rücksetzen Ein schreibender Aufruf dieser Funktion beendet die Tabellennachführung und verwirft die programmierte Tabelle. Diese kann nun reprogrammiert werden.
3	BUFSIZE	integer r/w	Anzahl Tabellenstützpunkte lesen/schreiben Mit dieser Funktion kann ein Datenpuffer für eine Tabelle reserviert werden. Die Größe kann nachträglich nicht mehr erhöht werden. Parameter / Rückgabewert ist die maximale Anzahl der Stützpunkte. Ein Tabellenstützpunkt besteht aus zwei Koordinatenwerten (für Master- und Follow-Achse). Bei Multi-Line-Tabelle muß BUFSIZE alle Zeilen einer Tabelle beinhalten. Diese Puffergröße wird verwendet, um den Speicher für mehr als eine Tabelle zu reservieren und <b>muß</b> einmalig gesetzt werden.
4	AXIS	integer r/w	Achs-Nr. der Achse, die mit dieser Tabelle nachgeführt werden soll lesen / schreiben
5	MASTER	integer r/w	Achs-Nr. der MasterAchse (Führungssachse) für diese Tabelle lesen / schreiben
6	GAIN	double r/w	Gain der Nachführachse lesen / schreiben Verstärkungsfaktor für die Tabellenstützpunkte der Nachführachse Defaultwert: 1.0
7	PHASE	double r/w	Phase der Führungsgröße lesen / schreiben Wert für die Verschiebung der Tabellenstützpunkte der Führungssachse Defaultwert: 0.0
8	SHIFT	double r/w	Shift der Nachführachse lesen / schreiben Wert für die Verschiebung der Tabellenstützpunkte der Nachführachse Defaultwert: 0.0
10	MODE	integer r/w	Betriebsart lesen / schreiben Durch das Beschreiben dieses Registers kann der Nachführmodus eingeschaltet werden. Weiterhin kann die Nachführung parametrierbar werden. Die Bitcodierung dieses Registers ist in Tabelle 4 dokumentiert. Bei jedem Schreibvorgang auf dieses Register müssen die aktuell gesetzten Bits übertragen werden.
11	ADDMASTER	double w	Stützpunkt in Mastertabelle eintragen SIZEMASTER wird dabei erhöht. Die Stützpunktwerte müssen stetig steigend (Ausnahme: erster Wert in einer Zeile bei Multi-Line-Tabellen), aber nicht äquidistant sein.
12	ADDFOLLOW	double w	Stützpunkt in Followtabelle eintragen SIZEFOLLOW wird dabei erhöht.
13	SIZEMASTER	integer r	Anzahl der Stützpunkte in Mastertabelle lesen
14	SIZEFOLLOW	integer r	Anzahl der Stützpunkte in Followtabelle lesen. In Mastertabelle und Followtabelle müssen gleich viele Werte eingetragen werden.
15	MLAXIS	integer r/w	Achse für Zeilenvorgabe lesen / schreiben (Multi-Line-Tabellen)

Tabelle 2: Funktionen ELCAM-Modul (Fortsetzung)

Nr.	Bez.	Typ	Erläuterung
16	MLSTART	double r/w	Startwert der Zeilenachse lesen / schreiben (Multi-Line-Tabellen)
17	MLEND	double r/w	Endwert der Zeilenachse lesen / schreiben (Multi-Line-Tabellen)
18	MLCOUNT	integer r/w	Anzahl der Zeilen lesen / schreiben (Multi-Line-Tabellen) mit diesem Wert wird die Multi-Line-Funktionalität aktiviert Defaultwert: 0
19	CALC TARGET POSITION	double r	Mit Hilfe dieser Funktion kann, bei ausgeschaltetem Nachführmodus, die derzeitige Position der Nachführachse über die Tabelle berechnete werden. Somit ist es möglich, vor dem aktivieren des Nachführmodus, die Startposition anzufahren und somit unkontrollierte Sprünge der Achse zu vermeiden. Der Aufruf sollte bei stehender Führungsachse erfolgen. Falls der Nachführmodus während des Aufrufes bereits aktiviert ist, liefert die Funktion rdOptionDbl den Wert BUSY (2) zurück. (Funktion verfügbar ab OS-Version V2.5.3.25)
20	ReadEICam Memory Address	integer r	Lesen der Speicheradresse der EICam Tabelle. Diese Funktion wird intern verwendet und wird in Applikationsprogrammen nicht benötigt.
21	SetTabSize	integer w	TabSizeMaster und TabSizeFollow setzen. Diese Funktion wird nur bei Ausführung aus dem Arbeitsspeicher heraus benötigt.
30	Axis Compensation Mode	integer r/w	DeviceNr = 0 / Kommando 30 in <b>Index</b> : Spindel-Kompensations-Modus aktivieren / deaktivieren mit 1 / 0 (gilt für alle Tabellen) DeviceNr = 1: Kommando 30 in <b>SubIndex</b> : Tabelle als Satz für Spindelsteigungsfehler-Kompensation markieren  Der übergebene Parameter enthält einen booleschen Wert, welche angibt ob die Funktionalität aktiviert oder deaktiviert wird. Mit dem Wert 0 wird die jeweilige Funktionalität abgeschaltet.  <b>Vorsicht:</b> Funktion bei DeviceNr. 0 und DeviceNr. 1 nicht verwechseln (Ergänzende Infos siehe unten)
31	MultiLine Linear Interpolation	integer r/w	Durch Zuweisung eines Wertes = 0 oder ungleich 0 kann bei Multiline-Kompensationstabellen angegeben werden, ob die Stützpunktwerte zwischen den Kompensationszeilen interpoliert werden. Wenn der Wert 0 zugewiesen ist, wird keine Interpolation vorgenommen, dann wird die Kompensationszeile an den Bereichsgrenzen rigoros umgeschaltet. <b>Vorsicht:</b> Diese Information kann versehentlich durch das Beschreiben des Mode-Registers wieder abgelöscht werden, wenn dort Bit 6 nicht gesetzt ist.
300	EICamPhysM emAdr	integer r/w	Mit dieser Funktion kann eine Physische Speicheradresse an das ELCAM-Modul übergeben werden. Die Variable BUFSIZE sollte erst nach der Übergabe dieser Speicheradresse beschrieben werden, da es ansonsten nicht möglich ist Tabellen zu definieren die über die Speichergröße in RWMOS hinausreichen.

### 2.2.1 Ergänzende Infos zum Aktivierung des AxisCompensationMode

Bei den ELCAM Achsverbänden muss unterschieden werden zwischen normalem ELCAM-Modus, Spindelsteigungsfehlerkompensation (siehe Kapitel 3.1) und Winkelfehlerkompensation (siehe Kapitel 3.2). Im normalen ELCAM Modus ergibt sich die Position der nachzuführenden Achse aus der Abhängigkeit, die in der entsprechenden Tabelle hinterlegt ist. Hier ist es nicht mehr möglich die nachgeführte Achse mit Verfahrkommandos zu bedienen. Bei Kompensationstabellen wird die aus der Tabelle berechnete

Kompensation zur Sollposition hinzugefügt. Die entsprechenden Achsen können weiterhin wie normale Achsen verwendet werden, ihre Position wird aber zu jedem Zeitpunkt korrigiert.

Tabelle 3: Fehlerstatuswort *Error* des ELCAM-Moduls

<b>Bit Nr.</b>	<b>Bez.</b>	<b>Erläuterung</b>
0..7		reserviert
8	<b>MemErr</b>	Speicher reicht für die Tabelle nicht aus
9	<b>SizeErr</b>	Unterschiedliche Anzahl von Stützpunkten oder weniger als 2 Werte bei Master und Follow eingetragen
10	<b>SizeMasterErr</b>	Zu viele Stützpunktwerte für Master-Achse eingetragen
11	<b>SizeFollowErr</b>	Zu viele Stützpunktwerte für Follow-Achse eingetragen
12	<b>MasterErr</b>	Masterachse falsch definiert
13	<b>NotAscending</b>	Stützpunktwerte der Master-Achse nicht monoton steigend
14	<b>LineTableError</b>	Fehler bei der Definition von Multi-Line-Tabellen
15	<b>MasterLineError</b>	bei Multi-Line-Tabellen: Fehler bei der Initialisierung der Multi-Line-Achse
16	<b>MLTableSize Error</b>	bei Multi-Line-Tabellen: Anzahl der Tabellenelemente muß ein ganzzahliges Vielfaches der Anzahl von Multiline-Zeilen sein
17	<b>LineStartPos Error</b>	bei Multi-Line-Tabellen: Startwerte der Führungssachse in Multilinetabellen muß in allen Zeilen gleich sein
18	<b>LineEndPos Error</b>	bei Multi-Line-Tabellen: Endwerte der Führungssachse in Multilinetabellen muß in allen Zeilen gleich sein
19..31		reserviert

Tabelle 4: Konfigurationsregister *Mode* des ELCAM-Moduls

Bit Nr.	Bez.	Erläuterung
0	<b>Run</b>	Mit diesem Bit wird die Tabellennachführung aktiviert
1	<b>RpModeTable</b>	Mit diesem Bit kann die Tabellennachführung auf Istwertnachführung eingestellt werden. Default ist Sollwertnachführung. Falls bei Steppersystemen gleichzeitig das Bit AuxModeTable gesetzt ist, wird das AUX-Register als Istwert herangezogen. <b>Wichtig:</b> Damit das AUX-Register verwendet werden kann, muss der Merker "Use encoder for position feedback" auf der Registerkarte "Motor specific parameters" der Systemdaten bei der jeweiligen Achse auf Yes gesetzt sein.
2	<b>RpModeLine</b>	Mit diesem Bit kann die Ermittlung des Zeilenindex nach dem Istwert eingestellt werden. Defaultmäßig wird hierzu der Achsensollwert herangezogen. Falls bei Steppersystemen gleichzeitig das Bit AuxModeLine gesetzt ist, wird das AUX-Register als Istwert herangezogen.
3	<b>MultiDimTable</b>	Dieses Bit zeigt an, daß eine Multi-Line-Tabelle verwendet wird. Diese Bit wird vom System gesetzt und kann vom Benutzer nicht direkt gesetzt werden.
4	<b>SinglePrecision</b>	Mit diesem Bit kann für die RWMOS-interne Darstellung der Tabellenstützpunkte der Datentyp float (32-bit Gleitpunkt) gewählt werden. Gegenüber der Standard-Darstellung mit 64 bit Gleitpunkt wird für die Tabelle nur der halbe Speicherplatz benötigt. Dieses Bit darf nur geändert werden, wenn noch keine Tabellenstützpunkte definiert sind und solange die Anzahl der Tabellenstützpunkte (BUFSIZE) noch nicht gesetzt ist. Ansonsten wird der Fehler 10 hex zurückgeliefert.
5	<b>AxisComp Set</b>	Dieses Bit zeigt an, daß es sich hierbei um einen Satz für die Kompensation des Spindelsteigungsfehlers / Winkelfehlers handelt.
6	<b>MultiLineLinear Interpolation</b>	Dieses Bit kann auch mit der Funktion SubIndex 31 gesetzt werden.
7	<b>AuxModeTable</b>	Mit diesem Bit kann die Tabellennachführung auf das Auxiliary (aux) Register eingestellt werden. Hierzu muss gleichzeitig das Bit <b>RpModeTable</b> gesetzt sein. Weiterhin ist diese Funktionalität nur bei Stepperachsen (Führungssachse) möglich. Für eine vernünftige Normierung der Positionseinheiten muss i.A. zur Laufzeit der Wert gfaux der Führungssachse richtig gesetzt werden. (Dieses Bit ist erst verfügbar ab RWMOS V2.5.3.115)
8	<b>AuxModeLine</b>	Mit diesem Bit kann die Ermittlung des Zeilenindex auf das Auxiliary (aux) Register eingestellt werden. Hierzu muss gleichzeitig das Bit <b>RpModeLine</b> gesetzt sein. Weiterhin ist diese Funktionalität nur bei Stepperachsen (Führungssachse) möglich. Für eine vernünftige Normierung der Positionseinheiten muss i.A. zur Laufzeit der Wert gfaux der Führungssachse richtig gesetzt werden. (Dieses Bit ist erst verfügbar ab RWMOS V2.5.3.115)

**Hinweis zum Konfigurationsregister:** Während der Konfiguration des Systems sind ggf. einige dieser Bits vor der Aktivierung der Tabellennachführung zu initialisieren (z.B. SinglePrecision). Bei nachfolgenden Schreibvorgängen müssen diese Registerwerte erhalten bleiben z.B. beim Start der Tabellennachführung. Deshalb ist es u.U. sinnvoll, eine Schattenvariable dieses Registers im Applikationsprogramm zu führen und diese nach entsprechender Modifikation zu schreiben.

## 2.3 Multi-Line Tabellen

Zur Nutzung dieser Funktionalität muss zunächst eine Achse definiert werden, welche den Tabellenindex vorgibt (MLAXIS). Weiterhin muss die Startposition, welche den Index 0 repräsentiert mit der Funktion MasterLineStart und die Endposition, welche den Maximalindex repräsentiert mit der Funktion MasterLineEnd vorgegeben werden. Die Anzahl der Zeilen wird mit der Funktion MasterLineCount definiert. Mit dem Wert MasterLineCount = 0 (Defaultwert) ist diese Funktionalität deaktiviert.

Bei der Programmierung der Stützpunkte bei zweidimensionalen Tabellen werden die Stützpunktwerte zeilenweise nacheinander programmiert. Jede Zeile muß gleich viele Stützpunkte beinhalten, und den gleichen Start- und Endwert für die Führungssachse haben. Vor dem Eintragen der Stützpunkte, muss MasterLineCount definiert sein, sonst ist es nicht möglich, beim Eintragen der zweiten Zeile, wieder mit kleinen Werten für die Führungssachse zu beginnen. In diesem Fall wird das Fehlerbit *NotAscending* gesetzt. Beim Aktivieren des Nachführmodus werden diese Bedingungen geprüft. Im Fehlerfall wird im Statuswort das Fehlerbit LineTableError gesetzt. Nach dem Start der Nachführung kann die Zeilenanzahl nicht mehr geändert werden.

## 2.4 Fehlerabfrage von ELCAM

Zur Fehlerabfrage muss das Error-Register (SubIndex = 1) gelesen werden. Die Bitcodierung dieses Registers ist in Tabelle 3 beschrieben. Falls ein Fehler angezeigt wird, darf die Tabellennachführung nicht aktiviert werden, da ansonsten mit unkontrolliertem Verhalten gerechnet werden muss. Allerdings müssen die entsprechenden Konfigurationsbits im Mode Register während der Fehlerabfrage bereits gesetzt sein, da ansonsten Plausibilitätskonflikte auftreten können.

## **2.5 Weitere Eigenschaften der ELCAM Funktionalität**

### **2.5.1 Tabellenwiederholung**

Eine programmierte Tabelle wiederholt sich bei der Ausführung jeweils nach einem Durchlauf des gesamten Master-Verfahrbereiches. Wenn diese Wiederholung nicht gewünscht ist, kann diese durch programmieren eines externen Wertes ausserhalb des MasterVerfahrbereichs am Anfang und/oder am Ende der Tabelle ausser Kraft gesetzt werden.

Die Tabellenwiederholung wirkt sich insbesondere dann aus, wenn sich die Nachführwerte beim ersten und beim letzten Tabelleneintrag unterscheiden. In diesem Fall wirkt eine Gainänderung während der Nachführung immer nur „dynamisch“ ab der Änderungsposition.

### **2.5.2 Endgeschwindigkeit**

Wenn die Tabelleninterpolation einer nachgeführten Achse deaktiviert ist, während sich diese Achse in Bewegung befindet, bleibt die aktuelle Geschwindigkeit dieser Achse erhalten. Falls dies nicht gewünscht ist, muss explizit z.B. das Kommando Jog-Stop (js) an die jeweilige Achse geschickt werden.

### **2.5.3 Programmierung mehrerer Nachführtabellen**

Bei der Programmierung / Verwendung unterschiedlicher Tabellen zur gleichen Zeit, werden die Zugriffe durch den Index der Tabelle im Feld Index des jeweiligen ObjectDescriptor-Elements referenziert. Der Index einer Tabelle wird vom Benutzer festgelegt.

Ab RWMOS V2.5.3.64 ist es auch möglich, für eine Nachführachse mehrere Tabellen zu verwalten. Das Aktivieren des Run-Modus einer Tabelle bewirkt automatisch eine Deaktivierung aller anderen Tabellen für die gleiche Achse. Somit ist es möglich, die Nachführung direkt von einer auf eine andere Tabelle umzuschalten. Die Umschaltung sollte jedoch in einem Verfahrbereich erfolgen, in dem die Nachführposition bei beiden Tabellen gleich ist, um unkontrollierte Achssprünge der Nachführachse bei der Umschaltung zu verhindern. Des weiteren ist es in diesem Fall nicht möglich mit Shift und Gain zu variieren, da auch dieses bei der Umschaltung zu Sprüngen der Achse führen würde.

## 2.6 Verwendung von ELCAM

- Vergabe eines Index und Zuweisung einer Master- und Slave-Achse
- Zuweisung der Tabellengröße
- Multi-LineTabellen: Anzahl der Tabellenzeilen definieren
- Multi-LineTabellen: Weitere Multi-Line Eigenschaften einstellen
- Eintragen der Master- und der Follow-Tabelle
- Status / Error abfragen
- Ideale Position der Nachführachse ermitteln und anfahren
- Nachführung aktivieren per Mode-Register  
Bit 0 = Nachführung ein  
Bit 1 = Istwertnachführung
- Führungssachse starten

## 2.7 Übertragung kompletter Tabellen per PCI-Direktzugriff

Bei sehr großen Tabellen (mehrere Megabyte), kann das Programmieren einer Tabelle mehrere Minuten in Anspruch nehmen. Um diese Ladezeiten zu vermeiden, ist es ab RWMOS.ELF V2.5.3.67 (mcug3.dll V2.5.3.43) möglich, eine Tabelle im PC Arbeitsspeicher aufzubereiten und mit einem einzigen Funktionsaufruf in den Arbeitsspeicher der Steuerung zu übertragen. Dadurch läßt sich die Ladezeit auf den Sekundenbereich verkürzen. Hierbei muß man unterscheiden, ob die interne Zahlendarstellung in 64bit-Gleitpunktzahlen oder in 32bit-Gleitpunktzahlen erfolgt (siehe hierzu Bit 4 im Konfigurationsregister Mode des EICAM-Modul).

Die Übertragung der Tabelle vom PC-Arbeitsspeicher in den Arbeitsspeicher der Steuerung erfolgt mit dem Kommando *wrEICamTable64* bei 64 bit Gleitpunktdarstellung und mit dem Kommando *wrEICamTable32* bei 32 bit Gleitpunktdarstellung.

Der Rückgabewert dieser Funktionen muß auf Erfolg abgeprüft werden. Bei erfolgreichem Laden der Tabelle wird der Wert 0 zurückgegeben. Werte ungleich 0 zeigen einen Fehler lt. nachfolgender Tabelle an.

Rückgabewert	Beschreibung
0	Funktion erfolgreich ausgeführt
7	Parameter size muß ein ganzzahliges Vielfaches von 8 bei 32-bit Gleitpunktdarstellung bzw. von 16 bei 64-bit Gleitpunktdarstellung sein
1	Die EICam-Tabelle ist bereits aktiviert oder der Funktionsaufruf wurde nicht entsprechend Bit 4 (float oder double) des EICam-Status-Registers gewählt; <i>oder</i> der Zugriff auf des EICAM-Object-Interface ist nicht möglich.
2	Der Parameter size ist zu groß oder zu klein.
3	Systemfehler, z.B. wenn das System während der Programmausführung von einer anderen Applikation neu gebootet wurde.
4	Adresse des EICAM-Speichers kann nicht ermittelt werden, z.B. weil RWMOS diese Funktion noch nicht unterstützt (erst ab V2.5.3.67)
5	DLL interner Speicherzugriff nicht möglich, EICAM-Speicher ist zu groß, oder die Datei mcug3.dll muß angepasst werden
6	Systemfehler, z.B. wenn das System während der

Programmausführung von einer anderen Applikation neu gebootet wurde.
--

Bei Verwendung dieser Methoden wird nicht überprüft ob die X-Werte monoton steigend sind. Auch die Vollständigkeit der Tabelle kann hier nicht überprüft werden.

Zur Verwendung dieser Funktion wird zunächst die Initialisierung des EICAM Moduls in gewohnter Weise durchgeführt.

- Vergabe eines Index und Zuweisung einer Master- und Slave-Achse
- ggf. Setzen des Bits SinglePrecision im Register Mode
- Zuweisung der Tabellengröße
- Multi-LineTabellen: Anzahl der Tabellenzeilen definieren
- Multi-LineTabellen: Weitere Multi-Line Eigenschaften einstellen

Danach wird anstatt der Aufrufe von ADDMASTER und ADDFOLLOW jeweils x und y einer Tabelle lt. dem untenstehendem Aufbau im PC-Arbeitsspeicher beschrieben. Nachdem die Tabelle vollständig beschrieben ist, wird diese mit einem der o.g. Kommandos an die Steuerung übertragen. Nach der Übertragung der Tabelle folgen die Kommandos zur Aktivierung der Tabelle in gewohnter Weise:

- Status / Error abfragen
- Ideale Position der Nachführachse ermitteln und anfahren
- Nachführung aktivieren per Mode-Register  
Bit 0 = Nachführung ein  
Bit 1 = Istwertnachführung
- Führungssachse starten

### 2.7.1 Aufbau der EICAM-Tabelle mit 64bit Gleitpunktdarstellung

double x0	double y0
double x1	double y1
double x2	double y2
...	...
double xn	double yn

Beispiel in C:

```
struct {
    double x, y;
} EICamTable[TABLESIZE];
```

Beispiel in Delphi:

```
TABLEPOINT = record
    x : double;
    y : double;
end;
ELCAMTABLE = ARRAY [0..TABLESIZE-1] of TABLEPOINT;
```

## 2.7.2 Aufbau der EICAM-Tabelle mit 32bit Gleitpunktdarstellung (Single Precision)

float x0	float y0
float x1	float y1
float x2	float y2
...	...
float xn	float yn

Beispiel in C:

```
struct {
    float x, y;
} EICamTable[TABLESIZE];
```

Beispiel in Delphi:

```
TABLEPOINT = record
    x : single;
    y : single;
end;
ELCAMTABLE = ARRAY [0..TABLESIZE-1] of TABLEPOINT;
```

## 2.8 Ausführung von ELCAM-Tabellen im PC-Arbeitsspeicher

Um mit größeren Tabellen arbeiten zu können, als der Arbeitsspeicher der MCU-3000 aufnehmen kann, wurde ab RWMOS V2.5.3.68 die Möglichkeit geschaffen, eine ELCAM-Tabelle im PC-Arbeitsspeicher zu erstellen und diese direkt per PCI-Memoryzugriffen zu verwenden. Dadurch entfällt die Übertragung auf die Steuerung. Hierbei ist jedoch zu beachten, daß die Tabelle in sogenanntem "physischen Speicher" abgelegt werden muß. Dieser Speicher hat die Eigenschaft, daß er über den gesamten Bereich fortlaufend allokiert ist und daß die "physische Speicheradresse" bekannt ist. Bei gewöhnlichen Datenobjekten in PC-Programmen wird stets nur mit "virtuellen Adressen" gearbeitet.

Physischer Speicher kann mit der Funktion `allocPhysMem()` allokiert werden. Dies ist eine Funktion von `mcug3.dll`. Der Prototyp ist in den Hochspracheninterfaces der MCUG3-Toolset Software deklariert.

### 2.8.1 PCAP-Funktion `allocPhysMem`

<b>BESCHREIBUNG:</b>	Mit dieser Funktion kann physischer Speicher vom Windows Betriebssystem angefordert (allokiert) werden.
<b>BORLAND DELPHI:</b>	<code>function allocPhysMem (var VirtualAdr: Pointer; var PhysAdr: integer; size : Integer): integer;</code>
<b>C:</b>	<code>unsigned allocPhysMem (void **VirtualAdr, unsigned *PhysAdr, unsigned size);</code>
<b>PARAMETER:</b>	VirtualAdr: Zeiger auf die Virtuelle Adresse, diese muß im Delphi-Programm verwendet werden um den Speicher zu verwenden. PhysAdr: Platzhalter für die Physische Speicheradresse size: Größe des anzufordernden Speichers in bytes
<b>RÜCKGABEWERT:</b>	0 bei Erfolg, Fehlercode bei Mißerfolg

<b>ANMERKUNG:</b>	Der Erfolg dieser Funktion muß unbedingt überprüft werden. Speicherbereiche, die mit dieser Funktion allokiert wurden, müssen vor dem Beenden des Programms mit freePhysMem wieder freigegeben werden.
-------------------	--

## 2.8.2 PCAP-Funktion freePhysMem

<b>BESCHREIBUNG:</b>	Mit dieser Funktion kann zuvor allokiertes physisches Speicher wieder freigegeben werden.
<b>BORLAND DELPHI:</b>	function freePhysMem (VirtualAdr: Pointer): integer;
<b>C:</b>	unsigned freePhysMem (void **VirtualAdr);
<b>PARAMETER:</b>	VirtualAdr: Zeiger auf die Virtuelle Adresse, welche von allocPhysMem geliefert wurde.
<b>RÜCKGABEWERT:</b>	0 bei Erfolg
<b>ANMERKUNG:</b>	

## 2.8.3 Hinweise zur Verwendung von Physischem Speicher

Da Windows den Speicherbereich während der Laufzeit mehr oder weniger fragmentiert, ist es nicht immer möglich den angeforderten Speicher auch tatsächlich zu erhalten. Bei Verwendung dieser Funktion ist vorzugsweise Windows XP einzusetzen. Weiterhin ist es vorteilhaft den Speicher möglichst bald nach dem Booten des PC anzufordern, da jeder Aufruf eines Programms unter Windows den Speicher mehr fragmentiert.

Es ist auch durchaus möglich, daß der Speicher beim ersten Aufruf einer Funktion bereitgestellt wird, nach wiederholtem Start aber nicht mehr zur Verfügung steht.

Desweiteren gibt es eine Möglichkeit mit dem Miniporttreiber Version 8.00e, Physischen Speicher in der Bootphase für die erstmalige Verwendung zu reservieren. Mit dieser Methode wurden auf einem Versuchsrechner mit einem Windows XP System mit 1GB Speicher einmalig 512 MByte Speicher als Physischer Speicher zur Verfügung gestellt. Bei dieser Methode ist es notwendig, daß genau die Speichergröße mit allocPhysMem() angefordert wird, die zuvor reserviert wurde.

## 2.8.4 Verwendung der ELCAM-Tabelleninterpolation mit Physischem Speicher

Die Handhabung von ELCAM-Tabellen ist so durchzuführen wie in obigen Abschnitten beschrieben. Ergänzend dazu sind die nachfolgenden Punkte zu beachten.

- Verwendung von RWMOS.ELF ab V2.5.3.68
- Physischen Speicher erfolgreich allokiert
- Physische Speicheradresse an das ELCAM Modul übertragen mit Hilfe der Funktion 300
- Maximale Tabellengröße mit BUFSIZE definieren
- Aufbau einer Tabelle wie in Abschnitt 2.7.1 bzw. 2.7.2 beschrieben.
- Das Übertragen der Tabelle an die Steuerung entfällt, dafür muß die Größe der Tabelle mit der Variablen SetTabSize gesetzt werden. Hier muß die Anzahl der Tabellen-Stützpunkte übergeben werden.
- Initialisierung und Aktivierung der Tabelle wie oben beschrieben.
- Vor dem Beenden des Programms Physischen Speicher wieder freigegeben.

## 3 Spindelsteigungs- und Winkelfehlerkompensation

Mit Hilfe von Tabellen des ELCAM-Moduls kann eine Spindelsteigungsfehlerkompensation bzw. eine Winkelfehlerkompensation in rechtwinkligen Koordinatensystemen vorgenommen werden. Diese Funktionalität ermöglicht es, die Positioniergenauigkeit von Achssystemen durch eine Kompensation bekannter Fehler mit Hilfe von Ausgleichstabellen zu verbessern.

Die "Spindelsteigungs- und Winkelfehlerkompensation" ist nur verfügbar in RWMOS.ELF ab V2.5.3.68 und wenn die Option "optionELCAM" enthalten ist.

Der berechnete wirksame Kompensationswert der letztendlich bei einer Achse ist, wird bei der Istwerterfassung von der Istposition subtrahiert und ist somit für den Anwender nicht sichtbar. Falls dieser ermittelt oder angezeigt werden soll, kann dies ab RWMOS.ELF V2.5.3.108 mit Hilfe der Ressource # 73 gemacht werden. Diese Kompensationswert enthält die Summe aller wirksamen Kompensationstabellen (Steigungs- und Winkelfehler-Kompensation).

### 3.1 Spindelsteigungsfehlerkompensation

Bei dieser Kompensation wird ein etwaiger Fehler in der Spindelsteigung kompensiert. Hierzu wird eine Tabelle für die jeweilige Achse angelegt, bei welcher Master- und Slaveachse (Parameter MASTER und AXIS) die zu kompensierende Achse ist. Mit ADDMASTER werden die jeweiligen Stützpunktwerte auf der zu kompensierenden Achse in die Tabelle eingetragen. Mit der Funktion ADDFOLLOW wird der jeweilige Kompensationswert, also der auszugleichende Fehler eingetragen. Die Anzahl der Master- und Follow Werte muß gleich sein. Weiterhin muß die Masterkoordinate stetig steigend sein. Anfangs- und Endwert der Kompensationstabelle werden so gewählt, daß sie ausserhalb des Verfahrbereichs liegen. Positionswerte zwischen den Stützpunkten der Tabelle werden linear zwischen den benachbarten Tabellenstützpunkten interpoliert.

Durch Schreiben des Wertes "1" an die Variable "AxisCompensationMode" muss die Tabelle als Kompensationstabelle gekennzeichnet werden. Für jede Achse kann eine eigene Spindelsteigungsfehlerkompensationstabelle programmiert werden.

Weiterhin muss durch Schreiben auf DeviceNr = 0 und Index = 30 der SpindelKompensationsmodus aktiviert werden. Bei der Aktivierung dieser Tabelle(Subindex 10) muss im Parameter das Bit 5 (AxisCompSet) gesetzt sein.

### 3.2 Winkelfehlerkompensation

Positionierfehler, die durch mechanische Winkelfehler verursacht werden, können mit Hilfe einer Kompensationstabelle ebenfalls (wie in Kapitel 3.1) ausgeglichen werden. In diesem Fall sind die Indizes der Master und der Follow-Achse natürlich unterschiedlich. Für jede Achskombination kann eine eigene Tabelle programmiert werden. Ansonsten gelten alle Zusammenhänge wie in Kapitel 3.1 beschrieben.

Es ist auch möglich Multiline-Tabellen für die Winkelfehlerkompensation zu definieren. Somit kann die Kompensationstabelle in Abhängigkeit einer weiteren Achse ausgewählt werden, d.h. die Tabellenzeile wird geändert in Abhängigkeit einer weiteren Achse. Mit der Funktion 31 kann in diesem Modus zusätzlich noch angewählt werden, daß der Kompensationswert zwischen den Zeilen linear interpoliert berechnet wird.

Durch Schreiben des Wertes "1" an die Variable "AxisCompensationMode" muss die Tabelle als Kompensationstabelle gekennzeichnet werden.