
POSITIONIER- UND BAHNSTEUERUNG APCI-8001 und APCI-8008

ELCAM-Interface

Stand: 02.05.2017, ab Disk V2.53T
Rev. 11/112018

www.addi-data.de

1 Einführung	5
2 Verwendung der ELCAM-Funktion.....	6
2.1 Initialisierung von ELCAM	6
2.2 Funktionen des ELCAM-Moduls	7
2.2.1 Ergänzende Infos zur Aktivierung des AxisCompensationMode	8
2.3 Multi-Line-Tabellen.....	10
2.4 Fehlerabfrage von ELCAM.....	10
2.5 Weitere Eigenschaften der ELCAM-Funktion	11
2.5.1 Tabellenwiederholung	11
2.5.2 Endgeschwindigkeit.....	11
2.5.3 Programmierung mehrerer Nachführtabellen.....	11
2.6 Verwendung von ELCAM.....	11
2.7 Übertragung kompletter Tabellen per PCI-Direktzugriff.....	12
2.7.1 Aufbau der ELCAM-Tabelle mit 64-Bit-Gleitpunktdarstellung.....	13
2.7.2 Aufbau der ELCAM-Tabelle mit 32-Bit-Gleitpunktdarstellung (Single Precision)	13
2.8 Ausführung von ELCAM-Tabellen im PC-Arbeitsspeicher	14
2.8.1 PCAP-Funktion allocPhysMem	14
2.8.2 PCAP-Funktion freePhysMem	14
2.8.3 Hinweise zur Verwendung von physischem Speicher	15
2.8.4 Verwendung der ELCAM-Tabelleninterpolation mit physischem Speicher	15
3 Spindelsteigungs- und Winkelfehlerkompensation	16
3.1 Spindelsteigungsfehler-Kompensation	16
3.2 Winkelfehlerkompensation	16

1 Einführung

Die ELCAM-Funktion der Karten APCI-8001 und APCI-8008 ermöglicht es, einzelne Achsen des Systems über eine Tabellenfunktion einer Führungssachse 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-Funktion wird über das Universelle Object-Interface von ADDIPOS konfiguriert. ELCAM ist eine Option der Betriebssystemsoftware RWMOS.ELF. Mit dem Dienstprogramm fwsetup kann bei gebootetem System geprüft werden, ob das vorliegende RWMOS.ELF die Option unterstützt.

Standardmäßig steht für die ELCAM-Tabellen ein Speicherbereich von 100.000 Byte 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-Funktion

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

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

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

2.2 Funktionen des ELCAM-Moduls

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

Nr.	Bezeichnung	Typ	Erläuterung
1	ERROR	integer r/w	Fehlerstatus lesen / rücksetzen Bit-Codierung 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 muss 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 Default-Wert: 1.0
7	PHASE	double r/w	Phase der Führungsgröße lesen / schreiben Wert für die Verschiebung der Tabellenstützpunkte der Führungssachse Default-Wert: 0.0
8	SHIFT	double r/w	Shift der Nachführachse lesen / schreiben Wert für die Verschiebung der Tabellenstützpunkte der Nachführachse Default-Wert: 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 Bit-Codierung 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 Follow-Tabelle eintragen SIZEFOLLOW wird dabei erhöht.
13	SIZEMASTER	integer r	Anzahl der Stützpunkte in Mastertabelle lesen
14	SIZEFOLLOW	integer r	Anzahl der Stützpunkte in Follow-Tabelle lesen. In Mastertabelle und Follow-Tabelle müssen gleich viele Werte eingetragen werden.
15	MLAXIS	integer r/w	Achse für Zeilenvorgabe lesen / schreiben (Multi-Line-Tabellen)
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 Default-Wert: 0

Nr.	Bezeichnung	Typ	Erläuterung
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 Aufrufs bereits aktiviert ist, liefert die Funktion rdOptionDbl den Wert BUSY (2) zurück. (Funktion verfügbar ab OS-Version V2.5.3.25)
20	ReadElCam Memory Address	integer r	Lesen der Speicheradresse der ELCAM 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 Index : Spindel-Kompensations-Modus aktivieren / deaktivieren mit 1 / 0 (gilt für alle Tabellen) DeviceNr = 1: Kommando 30 in SubIndex :Tabelle als Satz für Spindelsteigungsfehler-Kompensation markieren Der übergebene Parameter enthält einen booleschen Wert, welcher angibt, ob die Funktionalität aktiviert oder deaktiviert wird. Mit dem Wert 0 wird die jeweilige Funktionalität abgeschaltet. Achtung: 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 Multi-Line-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. Achtung: Diese Information kann durch das Beschreiben des Mode-Registers versehentlich wieder gelöscht werden, wenn dort Bit 6 nicht gesetzt ist.
300	ElCamPhysMemAdr	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 zur Aktivierung des AxisCompensationMode

Bei den ELCAM-Achsverbünden muss unterschieden werden zwischen normalem ELCAM-Modus, Spindelsteigungsfehler-Kompensation (siehe Kapitel 3.1) und Winkelfehlerkompensation (siehe Kapitel 3.1). 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

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

Tabelle 4: Konfigurationsregister *Mode* des ELCAM-Moduls

Bit Nr.	Bezeichnung	Erläuterung
0	Run	Mit diesem Bit wird die Tabellennachführung aktiviert
1	RpModeTable	Mit diesem Bit kann die Tabellennachführung auf Ist-Wert-Nachführung eingestellt werden. Standard ist Soll-Wert-Nachführung. Falls bei Steppersystemen gleichzeitig das Bit AuxModeTable gesetzt ist, wird das AUX-Register als Istwert herangezogen. Wichtig: 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	RpModeLine	Mit diesem Bit kann die Ermittlung des Zeilenindex nach dem Ist-Wert eingestellt werden. Standardmäßig wird hierzu der Achsen-Soll-Wert herangezogen. Falls bei Steppersystemen gleichzeitig das Bit AuxModeLine gesetzt ist, wird das AUX-Register als Istwert herangezogen.
3	MultiDimTable	Dieses Bit zeigt an, dass eine Multi-Line-Tabelle verwendet wird. Dieses Bit wird vom System gesetzt und kann vom Benutzer nicht direkt gesetzt werden.
4	SinglePrecision	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	AxisComp Set	Dieses Bit zeigt an, dass es sich hierbei um einen Satz für die Kompensation des Spindelsteigungsfehlers / Winkelfehlers handelt.
6	MultiLineLinear Interpolation	Dieses Bit kann auch mit der Funktion SubIndex 31 gesetzt werden.

Bit Nr.	Bezeichnung	Erläuterung
7	AuxModeTable	Mit diesem Bit kann die Tabellennachführung auf das Auxiliary (aux) Register eingestellt werden. Hierzu muss gleichzeitig das Bit RpModeTable gesetzt sein. Diese Funktionalität ist nur bei Stepperachsen (Führungssachse) möglich. Für eine vernünftige Normierung der Positionseinheiten muss im Allgemeinen zur Laufzeit der Wert gfaux der Führungssachse richtig gesetzt werden. (Dieses Bit ist erst verfügbar ab RWMOS V2.5.3.115)
8	AuxModeLine	Mit diesem Bit kann die Ermittlung des Zeilenindex auf das Auxiliary (aux) Register eingestellt werden. Hierzu muss gleichzeitig das Bit RpModeLine gesetzt sein. Diese Funktionalität ist nur bei Stepperachsen (Führungssachse) möglich. Für eine vernünftige Normierung der Positionseinheiten muss im Allgemeinen 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 gegebenenfalls 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 unter Umständen 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 (Default-Wert) 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 Bit-Codierung 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-Funktion

2.5.1 Tabellenwiederholung

Eine programmierte Tabelle wiederholt sich bei der Ausführung jeweils nach einem Durchlauf des gesamten Master-Verfahrbereichs. Wenn diese Wiederholung nicht gewünscht ist, kann diese durch Programmieren eines extremen Werts außerhalb des Master-Verfahrbereichs 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 Gain-Änderung 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 z.B. das Kommando Jog-Stop (js) explizit an die jeweilige Achse geschickt werden.

2.5.3 Programmierung mehrerer Nachföhrtabellen

Bei der Programmierung / Verwendung unterschiedlicher Tabellen zur gleichen Zeit werden die Zugriffe durch den Index der Tabelle im Feld Index des jeweiligen Object-Descriptor-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-Line-Tabellen: Anzahl der Tabellenzeilen definieren
- Multi-Line-Tabellen: 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 = Ist-Wert-Nachführung
- Führungsschse 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 64-Bit-Gleitpunktzahlen oder in 32-Bit-Gleitpunktzahlen erfolgt (siehe hierzu Bit 4 im Konfigurationsregister Mode des ELCAM-Moduls).

Die Übertragung der Tabelle vom PC-Arbeitsspeicher in den Arbeitsspeicher der Steuerung erfolgt bei 64-Bit-Gleitpunktdarstellung mit dem Kommando *wrELCamTable64* und bei 32-Bit-Gleitpunktdarstellung mit dem Kommando *wrELCamTable32*.

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 laut 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 ELCAM-Tabelle ist bereits aktiviert oder der Funktionsaufruf wurde nicht entsprechend Bit 4 (float oder double) des ELCAM-Status-Registers gewählt. oder der Zugriff auf des ELCAM-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 ELCAM-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, ELCAM-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 ELCAM-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-Line-Tabellen: Anzahl der Tabellenzeilen definieren
- Multi-Line-Tabellen: weitere Multi-Line-Eigenschaften einstellen

Danach werden anstatt der Aufrufe von ADDMASTER und ADDFOLLOW jeweils x und y einer Tabelle laut dem unten stehenden Aufbau im PC-Arbeitsspeicher beschrieben. Nachdem die Tabelle vollständig beschrieben ist, wird diese mit einem der oben genannten 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 = Ist-Wert-Nachführung
- Führungssachse starten

2.7.1 Aufbau der ELCAM-Tabelle mit 64-Bit-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 ELCAM-Tabelle mit 32-Bit-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 Tabellen arbeiten zu können, die größer sind als der Arbeitsspeicher der APCI-8001, wurde ab RWMOS V2.5.3.68 die Möglichkeit geschaffen, eine ELCAM-Tabelle im PC-Arbeitsspeicher zu erstellen und diese direkt per PCI-Memory-Zugriffen zu verwenden. Dadurch entfällt die Übertragung auf die Steuerung. Hierbei ist jedoch zu beachten, dass die Tabelle im sogenannten „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 Hochsprachen-Interfaces der ADDIPOS-Toolset-Software deklariert.

2.8.1 PCAP-Funktion `allocPhysMem`

BESCHREIBUNG:	Mit dieser Funktion kann physischer Speicher vom Windows-Betriebssystem angefordert (allokiert) werden.
BORLAND DELPHI:	<code>function allocPhysMem (var VirtualAdr: Pointer; var PhysAdr: integer; size : Integer): integer;</code>
C:	<code>unsigned allocPhysMem (void **VirtualAdr, unsigned *PhysAdr, unsigned size);</code>
PARAMETER:	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 Byte
RÜCKGABEWERT:	0 bei Erfolg, Fehlercode bei Misserfolg
ANMERKUNG:	Der Erfolg dieser Funktion muss unbedingt überprüft werden. Speicherbereiche, die mit dieser Funktion allokiert wurden, müssen vor dem Beenden des Programms mit <code>freePhysMem</code> wieder freigegeben werden.

2.8.2 PCAP-Funktion `freePhysMem`

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

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.

Des Weiteren gibt es eine Möglichkeit, mit dem Miniport-Treiber 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 1 GB Speicher einmalig 512 MByte Speicher als physischer Speicher zur Verfügung gestellt. Bei dieser Methode ist es notwendig, dass genau die Speichermenge 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 allokalieren
- 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 freigeben.

3 Spindelsteigungs- und Winkelfehlerkompensation

Mit Hilfe von Tabellen des ELCAM-Moduls kann eine Spindelsteigungsfehler-Kompensation 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 Kompensationswert, der letztendlich bei einer Achse wirksam 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. Dieser Kompensationswert enthält die Summe aller wirksamen Kompensationstabellen (Steigungs- und Winkelfehler-Kompensation).

3.1 Spindelsteigungsfehler-Kompensation

Bei dieser Kompensation wird ein etwaiger Fehler in der Spindelsteigung kompensiert. Hierzu wird eine Tabelle für die jeweilige Achse angelegt, bei welcher Master- und Slave-Achse (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 Spindelsteigungsfehler-Kompensationstabelle programmiert werden.

Des Weiteren 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, Multi-Line-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 Werts „1“ an die Variable „AxisCompensationMode“ muss die Tabelle als Kompensationstabelle gekennzeichnet werden.