



PA – Software

PLC-1131-3

PLC - CNC - Funktions-Interface

PA 8000

AUSGABE

Software Revision

Copyright

IRRRTUM UND TECHNISCHE ÄNDERUNGEN VORBEHALTEN

10.01

1.9

PA

Inhalt

1 Einführung.....	1
1.1 Allgemein	1
1.2 Bibliotheken.....	1
1.2.1 Standardbibliothek	1
1.2.2 Die PLC– Bibliotheken	2
1.3 Funktionen	2
1.4 Funktionsblock.....	5
2 Bibliothek für den Datenaustausch zwischen PLC und CNC CNCPLC.LIB.....	7
2.1 Spindel Funktionen	7
2.1.1 WRITE_SWORD (FUN).....	7
2.1.2 WRITE_SPOTI (FUN).....	7
2.1.3 DEACTIVATE_SWORD (FUN).....	8
2.1.4 DEACTIVATE_SPOTI (FUN).....	8
2.1.5 WRITE_SPINDLE_ORIENTATION (FUN).....	8
2.1.6 ACTIVATE_PLC_SWORD (FUN).....	9
2.1.7 ACTIVATE_SPOTI (FUN).....	10
2.1.8 WRITE_SPINDLE_CODE FUN)	10
2.1.9 READ_SWORD (FB)	10
2.2 Handwheel Functions	11
2.2.1 HANDWHEEL_DIRECTION (FUN)	11
2.2.2 HANDWHEEL_FACTOR_AUTO1 (FUN)	11
2.2.3 HANDWHEEL_FACTOR_AUTO2 (FUN)	12
2.2.4 RESET_HANDWHEEL_FACTOR (FUN)	12
2.2.5 HANDWHEEL_FACTOR_MAN1 (FUN)	13
2.2.6 HANDWHEEL_FACTOR_MAN2 (FUN)	13
2.3 Analog Channel CNC-Axis	13
2.3.1 READ_ANALOG_CHANNEL (FB).....	13
2.3.2 WRITE_ANALOG_CHANNEL (FUN)	14
2.4 Tool Management.....	14
2.4.1 WRITE_TOOL_NUMBER (FUN)	14
2.4.2 WRITE_SPARE_TOOL (FUN).....	14

2.4.3 WRITE_GEOMETRIC_D (FUN)	14
2.4.4 WRITE_GEOMETRIC_H (FUN)	15
2.4.5 WRITE_GEOMETRIC_Z (FUN).....	15
2.4.6 WRITE_RATE_X (FUN)	15
2.4.7 WRITE_RATE_Y (FUN)	15
2.4.8 WRITE_SUPPLEM_X (FUN).....	16
2.4.9 WRITE_SUPPLEM_Y (FUN).....	16
2.4.10 WRITE_TOOL_TYPE (FUN)	16
2.4.11 WRITE_NO_PIECES (FUN).....	16
2.4.12 WRITE_PLANNED_TOOL_LIFE (FUN)	17
2.4.13 WRITE_REMAININGTOOL_LIFE (FUN).....	17
2.4.14 WRITE_TOOL_WARNING_LIMIT (FUN)	17
2.4.15 WRITE_BREAKING_PARAMETER (FUN).....	17
2.4.16 WRITE_CUTTING_FORCE (FUN)	18
2.4.17 WRITE_POSITION (FUN)	18
2.4.18 READ_TOOL_NUMBER (FB).....	18
2.4.19 READ_SPARE_TOOL (FB).....	18
2.4.20 READ_GEOMETRIC_D (FB).....	19
2.4.21 READ_GEOMETRIC_H (FB).....	19
2.4.22 READ_GEOMETRIC_Z (FB).....	19
2.4.23 READ_RATE_X (FB).....	19
2.4.24 READ_RATE_Y (FB).....	19
2.4.25 READ_SUPPLEM_X (FB)	20
2.4.26 READ_SUPPLEM_Y (FB)	20
2.4.27 READ_TOOL_TYPE (FB).....	20
2.4.28 READ_NO_PIECES (FB)	20
2.4.29 READ_PLANNED_TOOL_LIFE (FB).....	20
2.4.30 READ_REMAININGTOOL_LIFE (FB)	21
2.4.31 READ_TOOL_WARNING_LIMIT (FB).....	21
2.4.32 READ_BREAKING_PARAMETER (FB)	21
2.4.33 READ_CUTTING_FORCE (FB)	21
2.4.34 READ_POSITION (FB).....	22

2.5 Maschinenparameter	22
2.5.1 WRITE_APPL_DATA (FUN).....	22
2.5.2 READ_APPL_DATA (FB)	22
2.6 PLC-Meldungen im MMI.....	23
2.6.1 PLC Meldungsdatei im MMI.....	23
2.6.2 WRITE_ERROR_MESSAGE.....	24
2.6.3 CLEAR_ERROR_MESSAGES	25
2.7 NC Programm	25
2.7.1 LOAD_NC_PROGRAM (FB)	25
2.7.2 ERASE_NC_PROGRAM_MEMORY (FB).....	25
2.7.3 SELECT_NC_PROGRAM (FB)	26
2.7.4 Fehlernummern	26
2.7.5 Hinweise	27
2.7.6 READFIRSTBLOCK	28
2.7.7 SELECT_NC_STATION	28
2.8 Verfahren von Achsen	29
2.8.1 MOVE_POSITIONING_AXIS.....	29
2.8.2 SET_POSITIONING_FEED	29
2.8.3 SET_POSITIONING_ADJUST	30
2.8.4 MOVE_INTERPOLATING_AXIS	30
3 Bibliothek für allgemeine Datenübertragungen zwischen PLC und externen Einheiten.....	33
3.1 Übersicht.....	33
3.2 Beschreibung der Bibliotheksfunktionen	34
3.2.1 ExtOpenDataChannel.....	34
3.2.2 ExtReadData	36
3.2.3 ExtWriteData.....	37
3.2.4 ExtCloseDataChannel	38
3.2.5 ExtClearError	39
3.3 Hinweise.....	39
3.4 Beispiel	40
4 CNC Fehlerbehandlung in der PLC	45

4.1 Übersicht.....	45
4.2 Globale Variablen	45
4.3 Beschreibung des Funktionsblocks GET_CNC_ERR der CNCSTD Bibliothek.	46
4.4 Hinweise.....	49
4.5 Beispiel	50

1 Einführung

1.1 Allgemein

Der PLC verwendet ein Funktions - Interface zur Kommunikation mit der CNC. Dieses Funktions - Interface sind zusätzliche Bibliotheken.

1.2 Bibliotheken

Im Projekt können ein Reihe Bibliotheken enthalten sein, welche POU's, Datentypen, globale Variablen und benutzerspezifische Variablen verwenden können. Die Bibliothek "standard.lib" ist ein Standardteil des Programms und ist immer vorhanden.

Der Bibliothekmanager zeigt alle Bibliotheken, die mit dem aktuellen Projekt verbunden werden. Die POU's, die Datenarten und die globalen Variablen der Bibliotheken können in der gleiche Weise wie benutzerdefinierte POU's, Datenarten und globale Variablen verwendet werden. Der Bibliothekmanager kann mit dem Befehl "Window" "Bibliothekmanager" geöffnet werden.

1.2.1 Standardbibliothek

Die Bibliothek "standard.lib" ist immer vorhanden. Es enthält alle Funktionen und Funktion Blöcke, die durch die IEC1131-3 als Standard POU's für ein IEC - Programmiersystem gefordert werden. Der Unterschied zwischen einer Standardfunktion und einem Operator ist, daß der Operator implizit durch das Programmiersystem erkannt wird, während die Standard POU's an das Projekt (standard.lib) gebunden werden muß.

1.2.2 Die PLC– Bibliotheken

Die PLC– Bibliotheken sind Teil der PLC-1131-3. Diese Bibliotheken enthalten viele Funktionen und Funktionsblöcke für CNC.- und PLC-Anwendungen.

PLC– Bibliotheken:

- `cnclib.lib` Standard -Bibliothek
- `cncplc.lib` Bibliothek zur Kommunikation mit dem CNC-Kanal 1
- `cffplc.lib` Bibliothek zur Kommunikation mit dem CNC-Kanälen 1..8
- `extdata.lib` Bibliothek für allgemeine Datenübertragungen zwischen PLC und externen Einheiten

1.3 Funktionen

Eine Funktion ist ein POU, der genau ein Datenelement bringt (das aus einigen Elementen, wie Feldern oder Strukturen bestehen kann), wenn es verarbeitet wird, und dessen Aufruf in den Textsprachen als Operator in den Ausdrücken auftreten kann.

Wenn man eine Funktion deklariert, darf nicht vergessen werden, daß die Funktion einen Typbezeichner besitzen muß. Nach dem Funktionsnamen muß ein Doppelpunkt eingetragen werden, der von einem Typbezeichner gefolgt wird.

Eine korrekte Funktionsdeklaration kann wie dieses Beispiel aussehen:

```
FUNCTION Fct:INT;
```

Zusätzlich muß ein Resultat der Funktion zugewiesen werden. Das bedeutet, daß dieser Funktionsname als Zustandsgröße verwendet wird. Eine Funktionsdeklaration Erklärung fängt mit der Schlüsselwort `FUNKTION` an und endet mit `END_FUNCTION` .

Beispiel: Umwandlung eines BCD-Wertes in einen Integerwert

Die im Bearbeitungsprogramm programmierten Werte für die Spindeldrehzahl oder für die Werkzeugnummer werden im PLC-Interface als BCD-Werte übermittelt. Für die weitere Verwendung im PLC-Programm soll der BCD-Wert in eine Integerzahl gewandelt werden.

CNC

```
%
P000001
N10 ...
....
N90 T12 M6
N100 M3 S350
...
N9000 M2
```

Im Bearbeitungsprogramm wird über die Funktion T12 das Werkzeug Nummer 12 angefordert und mit S350 eine Spindeldrehzahl von 350 u/min gefordert.

PLC-Interface

ONBCDWORD1, 2 (dword 3,4) BCD-Wert

Die im Bearbeitungsprogramm geforderte Spindeldrehzahl bzw. die T-Nummer kann im PLC-Interface Dword 3,4 als BCD-Wert gelesen werden.

PLC-Program

```
IF ON_STROBE2 THEN
SpindleValueIN:=DECODE_BCD(BCD:=ONBCDWORD1);
END_IF

IF ON_STROBE3 THEN
ToolValueIN:=DECODE_BCD(BCD:=ONBCDWORD1);
END_IF
```

Ist das Signal ON_STROBE2 auf TRUE dann steht ein neuer S-Wert im PLC-Interface. Ist das Signal ON_STROBE3 auf TRUE dann steht ein neuer T-Wert im PLC-Interface. Diese BCD-Werte werden durch die Funktion DECODE_BCD in einen DINT-Wert umgewandelt und den Variablen SpindleValueIN bzw. ToolValueIN zugewiesen.

PLC-Funktion

FUNCTION DECODE_BCD : DINT

VAR

Units: DINT;

Tens: DINT;

Hundreds: DINT;

Thousands: DINT;

END_VAR

VAR_INPUT

BCD: DWORD;

END_VAR

Units := BCD AND 15;

Tens := ((BCD / 16) AND 15)*10;

Hundreds := ((BCD / 256) AND 15)*100;

Thousands := ((BCD / 4096) AND 15)*1000;

DECODE_BCD := (Units + Tens + Hundreds + Thousands);

Der Einfachheit halber ist die hier dargestellte Funktion auf vier Stellen begrenzt. Es ist natürlich jedem überlassen diese Funktion zu erweitern. Als interne Variablen sind die Stellen des BCD-Wertes bis Tausend definiert.

Als Eingangsvariable ist ein DWORD definiert um mit dem PLC-Interface kompatibel zu sein.

Die einzelnen BCD-Stellen werden ausgefiltert und mit der entsprechenden Wertigkeit multipliziert. Als Ergebnis der Funktion wird ein DINT-Wert zurückgemeldet.

1.4 Funktionsblock

Ein Funktionsblock ist ein POU, das ein oder mehr Werte während der Prozedur liefert. Im Vergleich mit einer Funktion stellt ein Funktionsblock keinen Rückgabewert zur Verfügung.

Eine Funktionsblockdeklaration beginnt mit dem Schlüsselwort `FUNCTION_BLOCK` und endet mit `END_FUNCTION_BLOCK`.

reproduktionen oder Instanzen (Kopien) eines Funktionsblockes können erstellt werden.

Jede Instanz besitzt eine eigene Kennung (den Instanznamen) und eine Datenstruktur, die seine Eingänge, Ausgänge und internen Variablen enthält. Instanzen werden lokal oder global als Variablen deklariert, während der Name des Funktionsblockes als Typ einer Kennung angezeigt wird.

Beispiel einer Instanz mit dem Namen `INSTANCE` des `FUB`-Funktionsblockes:

`INSTANCE: FUB;`

Example: Blinker

PLC-Program

```

PROGRAM Indicator
VAR
    Out_Indicator:BOOL;
    Indicator:BLINKER;
END_VAR

Indicator.T_on:=t#2s;
Indicator.T_off:=t#2s;

Indicator();
Out_Indicator:=Indicator.Signal;
  
```

Indicator ist ein FUB vom type BLINKER

kürzer,
Indicator(T_on:=t#2s,T_off:=t#2s);

PLC-Funktionsblock

```

FUNCTION_BLOCK BLINKER
  
```

```

VAR
    OffTimer: TP;
    OnTimer: TP;
END_VAR

VAR_INPUT
    T_on: TIME;
    T_off: TIME;
END_VAR

VAR_OUTPUT
    Signal: BOOL;
END_VAR

OnTimer(IN:=NOT(OffTimer.Q), PT:=T_on);
OffTimer(IN:=NOT(OnTimer.Q), PT:=T_off);
Signal:=OnTimer.Q;
  
```

Als interne FUB werden Off- und Ontimer definiert

Als Eingangsvariablen werden zwei Zeiten (TIME) für On und Off definiert.

Als Ausgang wird die Variable Signal definiert

Die Variable Signal arbeitet als Blinker

2 Bibliothek für den Datenaustausch zwischen PLC und CNC

CNCPLC.LIB

2.1 Spindel Funktionen

2.1.1 WRITE_SWORD (FUN)

Schreibt ein S-word für maximal 6 Spindeln in die CNC; dieses S-word überschreibt den vom Bediener im NC-Programm programmierten Wert.

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	SpindleNumber	Maske für Spindeln, die geschrieben werden sollen
	REAL	SWORD	S-word für Spindel

Ausgabedaten:

2.1.2 WRITE_SPOTI (FUN)

Schreibt einen Spindel-Override-Wert für die gewünschten Spindeln in die CNC.

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	SpindleNumber	Maske für Spindeln, die geschrieben werden sollen
	REAL	SWORD	Spindel-Poti für max. 6 Wellen

Ausgabedaten:

2.1.3 DEACTIVATE_SWORD (FUN)

Deaktiviert den zuvor geschriebenen S-word-Wert, so daß die programmierten Werte wieder aktiv werden.

Daten:	Typ	Name	Kommentar
--------	-----	------	-----------

Eingabedaten:	WORD	SpindleNumber	
---------------	------	---------------	--

Ausgabedaten:

2.1.4 DEACTIVATE_SPOTI (FUN)

Deaktiviert den Spindel-Override_wert von der PLC, so daß die Werte vom Bedienfeld gültig werden.

Daten:	Typ	Name	Kommentar
--------	-----	------	-----------

Eingabedaten:	WORD	SpindleNumber	
---------------	------	---------------	--

Ausgabedaten:

2.1.5 WRITE_SPINDLE_ORIENTATION (FUN)

Spindel Stop mit definierter Position.

Daten:	Typ	Name	Kommentar
--------	-----	------	-----------

Eingabedaten:	WORD	SpindleNumber	Spindelnummer (1 .. 6)
	REAL	SWORD	Position an der die Spindel positioniert (0.0 .. 360.0)

Ausgabedaten:	BOOL		Die Funktion gibt den Wert TRUE zurück, wenn das Kommando an die CNC gesendet wurde. Das bedeutet nicht, das es von der CNC ausgeführt wurde.
---------------	------	--	---

Hinweise

- Mit Aufruf von WRITE_SPINDLE_ORIENTATION kontrolliert die PLC die aktive Spindel, bis die Spindel die gewünschte Position erreicht hat. In den folgenden Fällen wird WRITE_SPINDLE_ORIENTATION inaktiv:
 - Emergency stop
 - Control Reset von der CNC
 - Aufruf von WRITE_SPINDLE_CODE, ACTIVATE_PLC_SWORD oder DEACTIVATE_SWORD von der PLC.
- Während WRITE_SPINDLE_ORIENTATION ist ein Satztransfer im CNC-Programm nicht möglich. Es ist sicherzustellen, dass sich die CNC im Zustand Stop oder Zyklus-Halt befindet, bevor die Funktion aktiviert wird. (siehe ON_CYCLEON, IN_TRANSF)
- Nur **eine** WRITE_SPINDLE_ORIENTATION kann aktiviert werden. Um mehr als eine Spindel zu positionieren, ist erst mit WRITE_SPINDLE_CODE ein Spindel-Stop Befehl an die aktive Spindel zu senden, bevor die nächste Spindel positioniert werden kann. Mit ON_STANDxx kann sichergestellt werden, ob die Spindel an der gesetzten Position gestoppt hat.
- Mit aufruf von DEACTIVATE_SWORD wird die Kontrolle über die Spindel an die CNC zurückgegeben. Ist noch ein CNC-Spindel-Befehl aktiv, wird der Befehl wieder ausgeführt.

2.1.6 ACTIVATE_PLC_SWORD (FUN)

Aktiviert den von der PLC geschriebenen S-Word-Wert.

Daten:	Typ	Name	Kommentar
--------	-----	------	-----------

Eingabedaten:	WORD	SpindleNumber	
---------------	------	---------------	--

Ausgabedaten:

2.1.7 ACTIVATE_SPOTI (FUN)

Aktiviert den von der PLC geschriebenen Spindel-Override-Wert.

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	SpindleNumber	
Ausgabedaten:			

2.1.8 WRITE_SPINDLE_CODE FUN)

Schreibt den Spindel-M-Code für bis zu 6 Spindeln gleichzeitig in die CNC. Mit diesem Befehl kann die Spindel ohne vorherige Programmierung über die CNC gestartet und gestoppt werden.

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	Spindle_Code	Spindel-Wort
Ausgabedaten:			

2.1.9 READ_SWORD (FB)

Liest die aktuellen Spindelausgaben der existierenden Spindeln

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	Spindle_number	
Ausgabedaten:	WORD	SWORD	

2.2 Handwheel Functions

2.2.1 HANDWHEEL_DIRECTION (FUN)

Aktiviert die Handrad-Funktion für die gewünschte Achse. Achsen, die einem Handrad fest zugeordnet sind, werden nicht beeinflusst.

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	Axis_number	
	WORD	Factor	

Ausgabedaten:

2.2.2 HANDWHEEL_FACTOR_AUTO1 (FUN)

Festlegen des Faktors für die Handrad-Inkremente in der Betriebsart AUTOMATIC. Achsen, die einem Handrad fest zugeordnet sind, werden nicht beeinflusst.

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	HANDWHEEL_NUMBER	>0 .. Faktor für die Handrad-Inkremente
	WORD	FACTOR	

Ausgabedaten:

2.2.3 HANDWHEEL_FACTOR_AUTO2 (FUN)

Festlegen des Faktors für die Handrad-Inkmente in der Betriebsart AUTOMATIK durch Auswahl einer der Faktoren aus EXTRADVORSCHUB(3)

Achsen, die einem Handrad fest zugeordnet sind, werden nicht beeinflußt.

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	HANDWHEEL_NUMBER	>0 .. Index für EXTRADVORSCHUB
	WORD	FACTOR	

Ausgabedaten:

2.2.4 RESET_HANDWHEEL_FACTOR (FUN)

Rücksetzen des Faktors für die Handrad-Inkmente auf den vom MMI gesetzten Wert.

Daten:	Typ	Name	Kommentar
Eingabedaten:			
Ausgabedaten:			

2.2.5 HANDWHEEL_FACTOR_MAN1 (FUN)

Festlegen des Faktors für die Handrad-Inkremente in der manuellen Betriebsart. Achsen, die einem Handrad fest zugeordnet sind, werden nicht beeinflusst.

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	HANDWHEEL_NUMBER	>0 .. Faktor für die Handrad-Inkremente
	WORD	FACTOR	

Ausgabedaten:

2.2.6 HANDWHEEL_FACTOR_MAN2 (FUN)

Festlegen des Faktors für die Handrad-Inkremente in der manuellen Betriebsart. Achsen, die einem Handrad fest zugeordnet sind, werden nicht beeinflusst.

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	HANDWHEEL_NUMBER	>>0 .. Faktor für die Handrad-Inkremente
	WORD	FACTOR	

Ausgabedaten:

2.3 Analog Channel CNC-Axis

2.3.1 READ_ANALOG_CHANNEL (FB)

Lesen der an der NC angeschlossenen analogen Eingänge.

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	CHANNEL_NUMBER	Wählt die zu lesenden Kanäle
Ausgabedaten:	WORD	ANALOG_VALUE	

PLC - CNC - Funktions - Interface

2.3.2 WRITE_ANALOG_CHANNEL (FUN)

Schreiben der an der NC angeschlossenen analogen Ausgänge.

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	CHANNEL_NUMBER	Wählt die zu schreibenden Kanäle
	WORD	ANALOG_VALUE	

Ausgabedaten:

2.4 Tool Management

2.4.1 WRITE_TOOL_NUMBER (FUN)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	Wählt die zu schreibenden Kanäle
	string	Werkzeugnummer	

Ausgabedaten:

2.4.2 WRITE_SPARE_TOOL (FUN)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	Wählt die zu schreibenden Kanäle
	string	Ersatzwerkzeugnummer	

Ausgabedaten:

2.4.3 WRITE_GEOMETRIC_D (FUN)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	Wählt die zu schreibenden Kanäle
	REAL	Geometrische Kompensation D	

Ausgabedaten:

2.4.4 WRITE_GEOMETRIC_H (FUN)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
	REAL	Geometrische Kompensation H	
Ausgabedaten:			

2.4.5 WRITE_GEOMETRIC_Z (FUN)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
	REAL	Geometrische Kompensation Z	
Ausgabedaten:			

2.4.6 WRITE_RATE_X (FUN)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
	REAL	Rate von Verschleiß X	
Ausgabedaten:			

2.4.7 WRITE_RATE_Y (FUN)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
	REAL	Rate von Verschleiß Y	
Ausgabedaten:			

2.4.8 WRITE_SUPPLEM_X (FUN)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
	REAL	Zusätzliche Kompensation X	
Ausgabedaten:			

2.4.9 WRITE_SUPPLEM_Y (FUN)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
	REAL	Zusätzliche Kompensation Y	
Ausgabedaten:			

2.4.10 WRITE_TOOL_TYPE (FUN)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
	WORD	Werkzeugtyp	
Ausgabedaten:			

2.4.11 WRITE_NO_PIECES (FUN)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
	REAL	Restanzahl der Teile	
Ausgabedaten:			

2.4.12 WRITE_PLANNED_TOOL_LIFE (FUN)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
	REAL	Geplante Werkzeuglebensdauer	
Ausgabedaten:			

2.4.13 WRITE_REMAININGTOOL_LIFE (FUN)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
	REAL	Restliche Werkzeuglebensdauer	
Ausgabedaten:			

2.4.14 WRITE_TOOL_WARNING_LIMIT (FUN)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
	REAL	Werkzeuglebensdauer- Warngrenze	
Ausgabedaten:			

2.4.15 WRITE_BREAKING_PARAMETER (FUN)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
	REAL	Bruchparameter	
Ausgabedaten:			

2.4.16 WRITE_CUTTING_FORCE (FUN)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
	REAL	Schneidkraftparameter	
Ausgabedaten:			

2.4.17 WRITE_POSITION (FUN)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
	REAL	Position	
Ausgabedaten:			

2.4.18 READ_TOOL_NUMBER (FB)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
Ausgabedaten:	string	Werkzeugnummer	

2.4.19 READ_SPARE_TOOL (FB)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
Ausgabedaten:	string	Ersatzwerkzeugnummer	

2.4.20 READ_GEOMETRIC_D (FB)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
Ausgabedaten:	REAL	Geometrische Kompensation D	

2.4.21 READ_GEOMETRIC_H (FB)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
Ausgabedaten:	REAL	Geometrische Kompensation H	

2.4.22 READ_GEOMETRIC_Z (FB)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
Ausgabedaten:	REAL	Geometrische Kompensation Z	

2.4.23 READ_RATE_X (FB)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
Ausgabedaten:	REAL	Rate von Verschleiß X	

2.4.24 READ_RATE_Y (FB)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
Ausgabedaten:	REAL	Rate von Verschleiß Y	

2.4.25 READ_SUPPLEM_X (FB)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
Ausgabedaten:	REAL	Zusätzliche Kompensation X	

2.4.26 READ_SUPPLEM_Y (FB)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
Ausgabedaten:	REAL	Zusätzliche Kompensation Y	

2.4.27 READ_TOOL_TYPE (FB)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
Ausgabedaten:	WORD	Werkzeugtyp	

2.4.28 READ_NO_PIECES (FB)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
Ausgabedaten:	REAL	Restanzahl der Teile	

2.4.29 READ_PLANNED_TOOL_LIFE (FB)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
Ausgabedaten:	REAL	Geplante Werkzeuglebensdauer	

2.4.30 READ_REMAININGTOOL_LIFE (FB)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
Ausgabedaten:	REAL	Restliche Werkzeuglebensdauer	

2.4.31 READ_TOOL_WARNING_LIMIT (FB)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
Ausgabedaten:	REAL	Werkzeuglebensdauer- Warngrenze	

2.4.32 READ_BREAKING_PARAMETER (FB)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
Ausgabedaten:	REAL	Bruchparameter	

2.4.33 READ_CUTTING_FORCE (FB)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
Ausgabedaten:	REAL	Schneidkraftparameter	

2.4.34 READ_POSITION (FB)

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	MAGACINE_NUMBER	
Ausgabedaten:	REAL	Position	

2.5 Maschinenparameter

2.5.1 WRITE_APPL_DATA (FUN)

Daten:	Typ	Name	Kommentar
Eingabedaten:	string	Identifizier des zu schreibenden Datums	
	WORD	INDEX	
	REAL	Wert	
Ausgabedaten:			

2.5.2 READ_APPL_DATA (FB)

Daten:	Typ	Name	Kommentar
Eingabedaten:	string	Identifizier des zu schreibenden Datums	
	WORD	INDEX	
Ausgabedaten:		Wert	

2.6 PLC-Meldungen im MMI

Um PLC-Meldungen durch den MMI anzuzeigen, bietet die MMI- / PLC-Schnittstelle die folgende Kommunikationsmöglichkeit an.

2.6.1 PLC Meldungsdatei im MMI

Die PLC-Programmierer können eine oder mehrere sprachabhängigen Textdateien definieren, die **PLCMsgxxx.txt** benannt werden für ihre eigenen PLC-Meldungen. „xxx“ ist ein Platzhalter (nur für Unterlagen) für den Sprachnamen (d.h.: PLCMsgEngl.txt für Englisch, PLCMsgGer.txt für Deutsch...).

Diese Textdatei muß neben den Standardsprachdateien im MMI-Startverzeichnis archiviert werden. In dieser Datei können die PLC-Programmierer ihre Meldungen mit den entsprechenden Meldungsnummern definieren (von 501 bis 2999).

Das Dateiformat ist dasselbe wie in den Standardsprachdateien (d.h. MmiEngl.txt).

Format:

[PLC]

; dieses ist der Kapitelname der Datei

; Kommentar, kann in der ganzen Datei verwendet werden,

; nur mit dem Semicolon in erste Position

; leere Zeilen sind möglich

00000 = 409

; das ist die WINDOWS Sprachidentifikation

00600 = Druck ist zu hoch

; dieses ist eine Meldung, mit Meldungsnummer 600

00800 = undefinierter Wert in _

; Meldung mit einem Platzhalter, „_“ stellt den Platz für variablen Text dar

Die WINDOWS Sprachidentifikation (ids) kann folgendes sein:

407 Deutsch

409 US Englisch

419 Russisch

809 British Englisch

2.6.2 WRITE_ERROR_MESSAGE

Die Funktion WRITE_ERROR_MESSAGE überträgt die (Fehler) Meldung zum MMI mit dem folgenden Parametern:

Eingabedaten:

Typ	Name	Kommentar
WORD	ERROR_NUMBER	Dieses ist die (Fehler) Meldungsnummer aus der PLC-Meldungsdatei, diese Zahl erscheint in dem Ikon des CNC-Meldungsfeldes im MMI (nur Zahlen zwischen 501 und 2999 sind erlaubt)
WORD	STATUS	== 1, 2, 3 Meldung kann gelöscht werden mit QUIT von der Tastatur oder mit der Funktion: CLEAR_ERROR_MESSAGES == 4, 5 Meldung kann gelöscht werden mit der Funktion: CLEAR_ERROR_MESSAGES == 6 Meldung kann nicht gelöscht werden
STRING (64)	TEXT	Variabler Meldungstext (Platzhalter, ""_""") mit einem Maximum von 64 Zeichen (nur ein Platzhalter pro den Text erlaubt)

Hinweis:

- Wenn die ERROR_NUMBER in der PLC-Meldungsdatei nicht gefunden wurde, oder keine Datei existiert, wird nur der existierende TEXT angezeigt.

2.6.3 CLEAR_ERROR_MESSAGES

Die Funktion CLEAR_ERROR_MESSAGE löscht die (Fehler) Meldung, die von der PLC bis zum STATUS = 5 in das MMI geschrieben wird. Diese Funktion hat weder Eingangs noch Ausgangsparameter.

2.7 NC Programm

2.7.1 LOAD_NC_PROGRAM (FB)

Lädt NC Programm in den CNC-Speicher

Daten:	Typ	Name	Kommentar
Eingabedaten:	STRING(64)	FILENAME	
Ausgabedaten:	BOOL	DONE	
	WORD	ERROR	siehe Fehlernummern

2.7.2 ERASE_NC_PROGRAM_MEMORY (FB)

Löscht alle NC Programme im CNC-Speicher

Daten:	Typ	Name	Kommentar
Eingabedaten:	STRING(64)	FILENAME	Dummyname
Ausgabedaten:	BOOL	DONE	
	WORD	ERROR	siehe Fehlernummern

2.7.3 SELECT_NC_PROGRAM (FB)

Das PLC-Programm soll NC Programme auf Festplatte für die CNC-Bearbeitung auswählen. Alle CNC Programme auf allen zugänglichen Laufwerken können ausgewählt werden.

Durch Erstellen einer Instanz des Funktionsblocks SELECT_NC_PROGRAM und dem Aufruf dieser Instanz kann der PLC-Programmierer ein NC Programm für die Ausführung auswählen.

Daten:	Typ	Name	Kommentar
Eingabedaten:	STRING(64)	FILENAME	Generell muß der Dateiname alle Pfadbestandteile enthalten um die Datei lokalisieren zu können. Die Datei muß auf dem spezifizierten Pfad vorhanden sein. Platzhalter sind nicht erlaubt
Ausgabedaten:	BOOL	DONE	
	WORD	ERROR	siehe Fehlernummern

2.7.4 Fehlernummern

Der SELECT_NC_PROGRAM Funktionsblock sendet im Fehlerfall eine Fehlernummer, ERROR, zurück:

178	Programmanwahl nicht erlaubt
191	Programm nicht gefunden
192	Zyklus Ein aktiv
282	Programm in Bearbeitung
827	CNC nicht in AUTOMATIK
761	Fehler Datei nicht gefunden
762	Fehler Datei lesen

765	Speicherzuweisungsfehler
766	Satz nicht gefunden
767	Fehler Programmsyntax
796	Fehler Verzeichnis nicht gefunden
824	Zugriff auf Dateion abgelehnt
825	Zu viele Dateien geöffnet

2.7.5 Hinweise

Es wird die Anwahl von CNC Teileprogrammen auf der Festplatte und im internen Speicher der CNC unterstützt. Wenn eine Pfadangabe im Dateinamen vorhanden ist wird die Anwahl auf dem angegebenen Pfad auf der Festplatte ausgeführt. Ist keine Pfadangabe vorhanden wird die Anwahl im internen Speicher durchgeführt. Die Programmanwahl ist nur in den CNC Automatikbetriebsarten möglich.

Das Programm kann abgewählt werden, indem ein neues Programm über MMI oder PLC angewählt wird. CNTRL Reset deaktiviert **nicht** das Programm. SELECT_NC_PROGRAM und LOAD_NC_PROGRAM kann nicht gleichzeitig verwendet werden. Erst muß eine Funktion beendet sein bevor die nächste gestartet wird. Das heißt, das ein Programm angewählt werden kann und wenn die Anwahl beendet ist kann ein Programm geladen werden.

Beispiel:

Das folgende PLC-Programm wählt das NC-Programm P002000 im Verzeichnis C:\PADATA\NCPROG an:

```
PROGRAM PLC_PRG
VAR
    SELECT_NC : SELECT_NC_PROGRAM;
    Step : INT;
END_VAR
```

CASE Step OF

```
1: SELECT_NC(FILENAME:='C:\PADATA\NCPROG\IP002000.');
```

```
    IF SELECT_NC.DONE = TRUE THEN
```

```
        STEP := 10;
```

```
    END_IF;
```

```
10: (* processing the NEXT Steps *)
```

2.7.6 READFIRSTBLOCK

Ließt den ersten Satz eines CNC Programms

Daten:	Typ	Name	Kommentar
Ausgabedaten:	BOOL	VALID	
	STRING(64)	NCBLOCK	

2.7.7 SELECT_NC_STATION

Das PLC-Programm kann das MMI zu jeder möglichem NC-Station schalten.

Die Funktion bringt TRUE zurück, wenn der Befehl zum MMI gesendet ist, es gibt jedoch keine Bestätigung für die Ausführung. Andererseits ist es garantiert, das, wenn die Funktion einmal mit TRUE geantwortet hat wird die Station ausgewählt.

Die tatsächlich ausgewählte Station des MMI wird gezeigt in **.MMISTATION**.

Daten:	Typ	Name	Kommentar
Eingabedaten:	WORD	STATION	Nummer der CNC-Station

2.8 Verfahren von Achsen

2.8.1 MOVE_POSITIONING_AXIS

Verfährt die spezifizierte Positionierachse zur gewünschten Position in Maschinenkoordinaten. Als Geschwindigkeit wird der mit SET_POSITIONING_FEED gewählte Wert genutzt. Ist kein Wert gesetzt, verfährt die Achse mit der Maximalgeschwindigkeit.

Die Bewegung wird sofort ausgeführt, sobald eine Funktion aufgerufen wird. Wenn diese Funktion während der Bewegung der Achse aufgerufen wird, verfährt die Achse auf die neue Position, selbst wenn sie in entgegengesetzter Richtung zum aktuellen Endpunkt ist.

Daten:	Typ	Name	Kommentar
Eingabedaten:	STRING(2)	AXIS	Achsbuchstabe
	REAL	POSITION	Endposition

2.8.2 SET_POSITIONING_FEED

Setzt die Vorschubgeschwindigkeit für die spezifizierte Achse auf den während der Bewegung zu verwendenden Wert ein. Als Einschaltwert wird der maximale zulässige Vorschubwert verwendet.

Die Geschwindigkeit muß spezifiziert werden in 1000 int. Incr. / min

Daten:	Typ	Name	Kommentar
Eingabedaten:	STRING(2)	AXIS	Achsbuchstabe
	REAL	FEED	Geschwindigkeit

2.8.3 SET_POSITIONING_ADJUST

Das Koordinatensystem einer Positionierachse kann schnell justiert werden (setzen auf einen neuen Nullpunkt) durch eine PLC-Funktion. Bei Auftreten des Meßsignals vom Antrieb wird das Koordinatensystem der Achse umdefiniert.

Daten:	Typ	Name	Kommentar
Eingabedaten:	STRING(2)	AXIS	Achsbuchstabe
	REAL	MEASPOS	Position, auf die das Koordinatensystem eingestellt wird in interne Inkremente.

2.8.4 MOVE_INTERPOLATING_AXIS

Diese Funktion erlaubt beide Arten von Achsen von der PLC aus zu positionieren oder interpolieren. Der Hauptzweck dieser Funktion ist, den Werkzeugwechsel einfacher zu gestalten. Vorher mußte der Benutzer NC-Programme verwenden, um auf eine Wechsellposition zu fahren. Folglich gab es eine Notwendigkeit den NC- und PLC-Prozess zu synchronisieren, was nicht sehr bequem ist. Die neue Funktion kann den Werkzeugwechsel vollständig durch das PLC - Programm durchführen und braucht folglich kein NC-Programm.

Daten:	Typ	Name	Kommentar
Eingabedaten:	STRING(2)	AXIS	Achsbuchstabe
	REAL	POSITION	

Nur ein Wert für jede Achse ist möglich.

Ein Aufruf der Funktionen bedeutet nicht die sofortige Bewegung der Achse. Um die Bewegung zu aktivieren muß der Benutzer das entsprechende Signal IN_IPOMVMT setzen. (siehe IN_IPOMVMT (dword 16), Enable/disable Achsbewegung der Interpolationsachsen über die PLC)

3 Bibliothek für allgemeine Datenübertragungen zwischen PLC und externen Einheiten

3.1 Übersicht

Ein PLC-Programm kann eine Bibliothek `extdata.lib` einbinden, die eine bidirektionale Nicht-Echtzeit-Datenübertragung zwischen PLC-Programm und allen Peripheriegeräten realisiert, die durch eine Win32 API erreicht werden können.

Eine Datenquelle, die für Datenaustausch mit dem PLC-Programm verwendet werden kann, wird im folgenden als Kanal bezeichnet. Die PLC kann bis 8 geöffnete Kanäle hintereinander handhaben. Ein einzelner Transfer von Daten vom PLC-Programm wird als eine Übertragung bezeichnet. Die PLC kann 1 aktive Übertragung zu/von jeden geöffneten Kanal behandeln. Eine Übertragung kann aus einer willkürlichen Anzahl von Bytes oder Zeichen nach oder von einem Kanal bestehen. Der Benutzermodus, der alle Datenübertragung behandelt, läuft asynchrone zum PLC-Takt ab. Der PLC-Ablauf stellt Mittel für das Überwachen des Zustandes und des Fortschritts irgendeiner begonnenen Übertragung zur Verfügung. Eine Übertragung kann Daten bezüglich seines Kanals entweder sequentiell oder als Randomzugriff adressieren. Die Einstellung seiner angeforderten Zugriffsrechte entsprechen den Zugriffsbeschränkungen des Kanals, die durch das OS eingestellt werden. Der PLC-Programmierer erhält Methoden, um genaue Fehlerinformationen zu erhalten und Fehlerzustände zurückzusetzen.

3.2 Beschreibung der Bibliotheksfunktionen

3.2.1 ExtOpenDataChannel

```
ExtOpenDataChannel  
  ( channel:DINT,  
    error: POINTER TO STRING(128),  
    typ:BYTE,  
    attr:BYTE,  
    name:POINTER TO STRING(256) ) : DINT;
```

channel : ist die Kanalnummer, die geöffnet werden soll (d.h. zugewiesen zu einer Datei oder Einheit), muß im Bereich 0,,7 sein.

error : ist ein Zeiger zu einer Zeichenkette (wo die Zeichenkette selbst nicht durch das PLC-Programm definiert werden darf), der auf eine Fehlermeldung zeigt, falls der Kanal einen Fehlerzustand verursacht hat

typ : bezeichnet die Art des Anschlusses

typ = 1: Disk-Files

Dieses schließt Remotedateien auf Computern im Netz mit ein. **name** muß der korrekte Pfadname sein, z.B. \\MYCOPUTER\C:\MYDIR\MYSUBDIR\MYFILE.DAT

typ = 2: Shared Memory Mapped-Files

Dieses ist für die Realisierung von Datenübertragungen zwischen Prozessen auf dem gleichen Computer über Shared memory notwendig. Der Zugriff ist vom RAM-Typ. Abweichende Einstellungen im **attr**-Parameter werden ignoriert.

typ = 3: Serielle Schnittstelle

Der Name muß COMx sein, x ist die Nummer einer vorhandenen und unbenutzten seriellen Schnittstelle auf dem lokalen Rechner. Sie hat immer sequentiellen Zugriff, trotz der Einstellungen in **attr**.

typ = 4: Drucker oder parallele Schnittstelle

Der Name muß PRN oder LPTx sein, x ist die Nummer einer vorhandenen und unbenutzten parallelen Schnittstelle auf dem lokalen Rechner. Sie hat immer sequentiellen Zugriff, trotz der Einstellungen in **attr**.

attr : ist ein Bitfeld, das jede mögliche Kombination der folgenden Flags enthalten kann (deshalb alle angeforderten Flag-Werte zusammenfügen):

Wert 1: FALSE=SEQ, TRUE=RAM

für RAM-Zugriff spezifiziert der Parameter *offset* der Lese-Schreibfunktion (sehen Sie unten), den Standort innerhalb der externen Datei, während SEQ Zugriff bedeutet, daß Daten angefügt/gelöscht werden am Ende der Datei/Buffers. Der Zugriff auf Memory-Dateien ist immer RAM, auf seriell und parallel Port immer SEQ. Für Diskdateien, der einfachste Modus (hängt von der Art der Anwendung ab) kann ausgewählt werden.

Wert 2: TRUE=MUST_EXIST, FALSE=WILL_BE_CREATED

trifft nur zu auf Dateien und Memory-Dateien. Spezifiziert ob die Datei, wenn nicht bereits existiert, hergestellt wird oder ob sie immer existieren muß, bevor sie für die Datenübertragung geöffnet werden können.

Wert 4: TRUE = NO_READ, FALSE = READ_PERMITTED

spezifiziert, ob Lesezugriff zum Kanal vom OS angefordert wird. Kann notwendig sein, um alle möglichen Zugriffsbeschränkungen durch den Eigentümer der Kanaldatei oder Einheit zu treffen.

Wert 8: TRUE = NO_WRITE, FALSE = WRITE_PERMITTED

spezifiziert, ob Schreibzugriff zum Kanal vom OS angefordert wird. Kann notwendig sein, um alle möglichen Zugriffsbeschränkungen durch den Eigentümer der Kanaldatei oder Einheit zu treffen.

name : ist der Pfad- oder Schnittstellenname für den Kanal, in Übereinstimmung mit der Windows NT Konventionen, die im PLC-Programm definiert werden muß.

3.2.2 ExtReadData

ExtReadData

```
( channel:DINT,  
  error: POINTER TO STRING(128),  
  data : POINTER TO BYTE,  
  attr:BYTE,  
  size : DINT,  
  offset : DINT ) : DINT;
```

channel : ist die Kanalnummer, die geöffnet werden soll, von der Daten eingelesen werden sollen (d.h. zugewiesen zu einer Datei oder Einheit), muß im Bereich 0,,7 sein.

- error :** ist ein Zeiger zu einer Zeichenkette (wo die Zeichenkette selbst nicht durch das PLC-Programm definiert werden darf), der auf eine Fehlermeldung zeigt, falls der Kanal einen Fehlerzustand verursacht hat
- data :** Die Daten sind in jedem möglichem Datenobjekt(en) oder - Array(s) völlig enthalten , die im PLC-Programm definiert werden.
- size :** Liest **size** Bytes von Daten vom geöffneten Kanal ein.
- offset :** Offset ist, bei Kanal für RAM-**type** , der Quelloffset und wird bei SEQ-**type** ignoriert .

3.2.3 ExtWriteData

ExtWriteData

(channel:DINT,
error: POINTER TO STRING(128),
data : POINTER TO BYTE,
size : DINT,
offset : DINT) : DINT;

- channel :** ist die Kanalnummer, die geöffnet werden soll, auf die Daten geschrieben werden sollen (d.h. zugewiesen zu einer Datei oder Einheit), muß im Bereich 0,,7 sein.
- error :** ist ein Zeiger zu einer Zeichenkette (wo die Zeichenkette selbst nicht durch das PLC-Programm definiert werden darf), der auf eine Fehlermeldung zeigt, falls der Kanal einen Fehlerzustand verursacht hat

- data :** Die Daten sind in jedem möglichem Datenobjekt(en) oder - Array(s) völlig enthalten , die im PLC-Programm definiert werden.
- size :** Schreibt **size** Bytes von Daten in den geöffneten Kanal ein.
- offset :** Offset ist, bei Kanal für RAM-**type** , der Zieloffset und wird bei SEQ-**type** ignoriert .

3.2.4 ExtCloseDataChannel

Diese Funktion schließt den entsprechenden Kanal nach Beenden aller Ein-/Ausgabeoperationen und gibt danach alle Recourcen frei. Der zu schließende Kanal sollte vor erfolgreich geöffnet worden sein, sonst entsteht ein Fehler.

ExtCloseDataChannel

(channel:DINT,
error: POINTER TO STRING(128)) : DINT;

- channel :** ist die Kanalnummer, die geschlossen werden soll (d.h. zugewiesen zu einer Datei oder Einheit), muß im Bereich 0,,7 sein.
- error :** ist ein Zeiger zu einer Zeichenkette (wo die Zeichenkette selbst nicht durch das PLC-Programm definiert werden darf), der auf eine Fehlermeldung zeigt, falls der Kanal einen Fehlerzustand verursacht hat.

3.2.5 ExtClearError

Wenn der gewählte Kanal geöffnet war bevor der Fehler auftrat, wird er wieder geöffnet und vorbereitet für neue Übertragungen. Wenn der Kanal geschlossen war (und der Fehler trat während eines Versuchs auf ihn zu öffnen), bleibt er geschlossen.

ExtClearError(DINT channel) : DINT;

channel : ist die Kanalnummer, in der ein möglicher Fehlerzustand zurückgesetzt wird (d.h. zugewiesen zu einer Datei oder Einheit), muß im Bereich 0..7 sein.

3.3 Hinweise

Der Return-Wert ist entweder 1 zum Anzeigen des Erfolges oder 0 für einen laufenden Antrag oder -1 für einen Fehler. Im letzten Fall zeigt **error** auf eine Meldung, die den Fehler beschreibt und **ExtClearError** kann verwendet werden, um den Fehlerzustand für diesen Kanal zurückzusetzen.

Die Funktion kann mit den gleichen Parametern wieder aufgerufen werden bis der Rückgabewertwert gleich 1 oder -1 ist.

Wenn die Funktion mit den gleichen Parametern aufgerufen wird und der Kanal bereits vorher geöffnet war, bringt die Funktion noch 1 zurück.

Es ist zu beachten, das eine anderer Aufruf der Funktion beginnt sobald die Funktion 1 für Erfolg zurückgemeldet hat (z.B. in der folgenden PLC-Schleife) !

Alle geöffneten Kanäle werden geleert und geschlossen, wenn das PLC-Programm vom Runtime-System oder vom Benutzer zurückgesetzt wird.

3.4 Beispiel

Das folgende Beispiel liest oder schreibt in eine Textdatei oder in eine binäre Datei.

```
PROGRAM PLC_MMI
```

```
VAR
```

```
    (* Variablen für lesen und schreiben der * txt Dateien*)
```

```
    File:          ARRAY [0..3] OF STRING(128);
```

```
    ErrPtr:        ARRAY [0..3] OF POINTER TO STRING(128);
```

```
    errormess:     STRING(128);
```

```
    OpenState:     ARRAY[0..3] OF DINT;
```

```
    CloseState:    ARRAY[0..3] OF DINT;
```

```
    ReadState:     ARRAY[0..3] OF DINT;
```

```
    WriteState:    ARRAY[0..3] OF DINT;
```

```
    (* Input- und Outputvariablen*)
```

```
    Read_1:        ARRAY[0..19] OF BYTE;
```

```
    Write_1:        ARRAY[0..31] OF BYTE;
```

```
    Read_2:         STRING(128);
```

```
    Write_2:        STRING(128);
```

```
    (*aktiver I/O Kanal*)
```

```
    channel:        INT;
```

```
    (* Groesse des File I/O *)
```

```
    size:           ARRAY[0..3] OF INT;
```

```
    (* Steps *)
```

```
    action:         ARRAY[0..3]OF INT:=0,0,0,0;
```

```
END_VAR
```

(* Definition der Dateinamen fuer I/O*)

```
File[0]:='C:\Files\User\PLC_Example_Read1.txt';
File[1]:='C:\Files\User\PLC_Example_Read2.txt';
File[2]:='C:\Files\User\PLC_Example_Write1.txt';
File[3]:='C:\Files\User\PLC_Example_Write2.txt';
```

(*Definition der Groesse fuer i/o Buffer *)

```
size[0]:=20;
size[1]:=32;
size[2]:=128;
size[3]:=128;
```

(* Schleife fuer File I/O *)

```
FOR channel:=0 TO 3 DO
  CASE action[channel] OF
    0:   OpenState[channel]:= ExtOpenDataChannel(channel,
          ADR(ErrPtr[channel]),1,1,
          ADR(File[channel]));
    IF (OpenState[channel] =-1) THEN
      errormess:=ErrPtr[channel]^;
      ExtClearError(channel);
      action[channel]:=99;
    ELSIF (OpenState[channel]=1) THEN
      IF (channel=0) THEN
        action[channel]:=1;
      ELSIF (channel=1) THEN
        action[channel]:=2;
      ELSIF (channel=2) THEN
        action[channel]:=3;
      ELSIF (channel=3) THEN
        action[channel]:=4;
      ELSE
        action[channel]:=99;
      END_IF;
    END_IF;
  END_IF;
```

```

1:  ReadState[channel]:=ExtReadData(channel,
      ADR(ErrPtr[channel]),ADR(Read_1),
      size[channel],0);
  IF (ReadState[channel]=-1) THEN
      errormess:=ErrPtr[channel]^;
      ExtClearError(channel);
      action[channel]:=99;
  ELSIF (ReadState[channel]=1) THEN
      action[channel]:=99;
  END_IF

2:  ReadState[channel]:=ExtReadData(channel,
      ADR(ErrPtr[channel]),ADR(Read_2),
      size[channel],1);
  IF (ReadState[channel]=-1) THEN
      errormess:=ErrPtr[channel]^;
      ExtClearError(channel);
      action[channel]:=99;
  ELSIF (ReadState[channel]=1) THEN
      action[channel]:=99;
  END_IF

3:  WriteState[channel]:=ExtWriteData(channel,
      ADR(ErrPtr[channel]),ADR(Write_1),
      size[channel],0);
  IF (WriteState[channel]=-1) THEN
      errormess:=ErrPtr[channel]^;
      ExtClearError(channel);
      action[channel]:=99;
  ELSIF (WriteState[channel]=1) THEN
      action[channel]:=99,*)
  END_IF

```



```

4:   WriteState[channel]:=ExtWriteData(channel,
        ADR(ErrPtr[channel]),ADR(Write_2),
        size[channel],0);
    IF (WriteState[channel]=-1) THEN
        errormess:=ErrPtr[channel]^;
        ExtClearError(channel);
        action[channel]:=99;
    ELSIF (WriteState[channel]=1) THEN
        action[channel]:=99;
    END_IF

99:   CloseState[channel]:=ExtCloseDataChannel(channel,
        ADR(ErrPtr[channel]));
    IF (CloseState[channel]=-1) THEN
        errormess:=ErrPtr[channel]^;
        ExtClearError(channel);
        action[channel]:=0;
    ELSIF (CloseState[channel]=1) THEN
        action[channel]:=0;
        OpenState[channel]:=0;
        CloseState[channel]:=0;
    END_IF;
END_CASE;
END_FOR;

```


4 CNC Fehlerbehandlung in der PLC

4.1 Übersicht

Alle Fehlermeldungen der CNC müssen ebenfalls in die PLC übertragen werden, wenn die Fehlermeldungen einem übergeordnetem System oder einer Visualisierungssoftware, z.B. einem PAVis-Projekt, zugänglich gemacht werden soll.

Mit globalen Variablen des PLC-Programms bekommt man die Information, wieviel Fehler sich in jeder Station (Kanal) ereignet haben.

Um mehr Informationen über die CNC-Fehler zu erhalten kann ein Funktionsblock aufgerufen werden. Dieser Funktionsblock liefert die folgenden Informationen eines CNC-Fehlers:

- Fehlernummer,
- Quittierungsstatus,
- Ereignis,
- Fehlertext in verschiedenen Sprachen und
- den CNC-Kanal (Station) des Fehlers.

4.2 Globale Variablen

Die globalen Variablen

- CNC_ERR_CNT,
- o2CNC_ERR_CNT,
- ...,
- o8CNC_ERR_CNT

im Merkerbereich der PLC (wie auch z.B. ON_STARTNC) informiert das PLC-Programm wieviel Fehler sich in der CNC ereignet haben (pro Kanal).

4.3 Beschreibung des Funktionsblocks GET_CNC_ERR der CNCSTD

Bibliothek

Durch Erstellen einer Instanz des Funktionsblocks GET_CNC_ERR und dem Aufruf dieser Instanz erhält der PLC-Programmierer alle CNC-Fehlerinformationen.

Der GET_CNC_ERR Funktionsblock hat drei Eingangsparameter:

REQ_CHANNEL: Der CNC-Kanal [1..8], von welchem die Information gelesen werden soll.

REQ_ERR_DIR: Flag für das Navigieren durch die CNC-Fehlerliste.

- 2 = erster CNC-Fehler
- 1 = vorhergehender CNC-Fehler
- 1 = nächster CNC-Fehler
- 2 = letzter CNC-Fehler.

Dieses Flag zeigt auf den ersten CNC-Fehler nach der Initialisierung des PLC-Programms.

LANG_ID: Die Sprach-ID in welcher der Fehlertext in die Ausgangsvariable ERR_TEXT abgelegt wird (siehe auch Ländercodes in der Windows Online Help und installierte MMI*.txt Files).

- 407 hex = 1031 = deutsch
- 409 hex = 1033 = US Englisch
- 419 hex = 1049 = Russisch

Nach Aufruf des Funktionsblocks informiert die Ausgangsvariable VALID das PLC-Programm, das alle Ausgangsvariablen komplett von der CNC gesetzt sind.

Folgende CNC-Fehlerinformationen sind verfügbar wenn die Variable VALID 1 ist:

ERR_NUMBER: Die CNC-Fehlernummer.

ERR_QUIT_STATE: Ob und wie dieser Fehler quittiert werden kann:

- 1 = Der nächste Tastendruck oder Paneleingabe löscht diesen Fehler
- 3 = Die CNC löscht den Fehler selbst
- 4 = Der Fehler kann durch drücken von Ctrl+X gelöscht werden (Quit)
- 5 = Der Fehler kann durch Ctrl+C (Reset) gelöscht werden
- 6 = Dieser Fehler kann nicht gelöscht werden. Nur ein Neustart der CNC ist möglich.

ERR_OCCASION: Dieser Wert zeigt den Fehlertyp:

- 0 = Start up Phase Hardwarefehler
- 1 = Start up Phase Systemfehler
- 2 = Hardwarefehler
- 3 = Systemfehler
- 4 = Servofehler
- 5 = PLC-Fehler
- 6 = Prozessfehler
- 7 = Computerfehler
- 8 = Anwenderspezifischer Fehler

- 9 = Cyclenfehler
- 10 = Bedienungsfehler
- 11 = Eingabefehler
- 12 = Kommunikationsfehler

ERR_TEXT: CNC-Fehler im Klartext. Siehe auch Eingabeparameter LANG_ID.

CHANNEL: Die CNC-Kanalnummer [1..8] des Fehlers.

VALID: Dieses Flag zeigt an, ob die Ausgangsvariablen gültig sind.

- 0 = Die Ausgangsvariablen dieses Funktionsblocks sind nicht gültig.
- 1 = Die Ausgangsvariablen dieses Funktionsblocks sind gültig.

STATE: Interne Information über den Zustand des aufgerufenen Funktionsblocks.

- 0 = Der Funktionsblock ist aktiv.
- 1 = Der Funktionsblock wurde aufgerufen und das PLC Runtime-System ist dabei die CNC Fehlerinformationen bereitzustellen.
- 10 = Die Anfrage wurde erfolgreich zur CNC gesendet.
- 20 = Die CNC antwortet und die PLC stellt die Informationen bereit.

4.4 Hinweise

Der gelesene CNC-Fehler wird aus der CNC-Fehlerliste gelöscht nachdem die Fehlerinformationen in die Variablen des Funktionsblocks transferiert wurden. Der Fehlerzähler des CNC-Kanals wird um eins dekrementiert.

Die Fehlertexte sind X in den MMI*.txt Dateien (Verzeichnis \program files\power automation\mmi) gespeichert. Bitte informieren Sie sich, welche MMI*.txt Dateien auf ihren System installiert sind. Die entsprechende Sprach-ID ist im Eintrag 00000 der MMI*.txt Datei abgelegt.

Es ist nicht nötig, den Aufruf des Funktionsblocks zu wiederholen bis das VALID Flag den Wert 1 erhält. Das PLC-Runtimesystem ruft weiterhin den Funktionsblock auf bis alle Informationen verfügbar sind.

Das PLC-Runtimesystem kann nur einen Aufruf dieses Funktionsblocks zur Zeit verarbeiten. Befinden sich mehrere Instanzen dieses Funktionsblocks im PLC-Programm oder es wird ein weiterer Aufruf gestartet während das PLC-Runtimesystem den ersten Aufruf bearbeitet, werden alle weiteren Aufrufe ignoriert. Eine Kontrolle kann mit der Ausgangsvariablen STATE erfolgen.

4.5 Beispiel

Das folgende PLC-Programm liest alle CNC-Fehler vom CNC-Kanal 1:

```
PROGRAM PLC_PRG
```

```
VAR
```

```
    NxtCncErr : BOOL := TRUE;
```

```
    CncErr : GET_CNC_ERR;
```

```
END_VAR
```

```
IF (NxtCncErr = TRUE) AND (CNC_ERR_CNT > 0) THEN
```

```
    CncErr(REQ_ERR_DIR:=1, REQ_CHANNEL:=1, LANG_ID:=1033);
```

```
END_IF
```

```
NxtCncErr := FALSE;
```

(* verhindert das lesen des naechsten
Fehlers bevor der vorhergehende
bearbeitet wurde *)

```
IF (CncErr.VALID = 1) THEN
```

```
    (* processing the CNC error *)
```

```
    NxtCncErr := TRUE;
```

(* setzen dieses Flag auf true,
nach Bearbeitung des CNC-Fehlers *)

```
END_IF
```