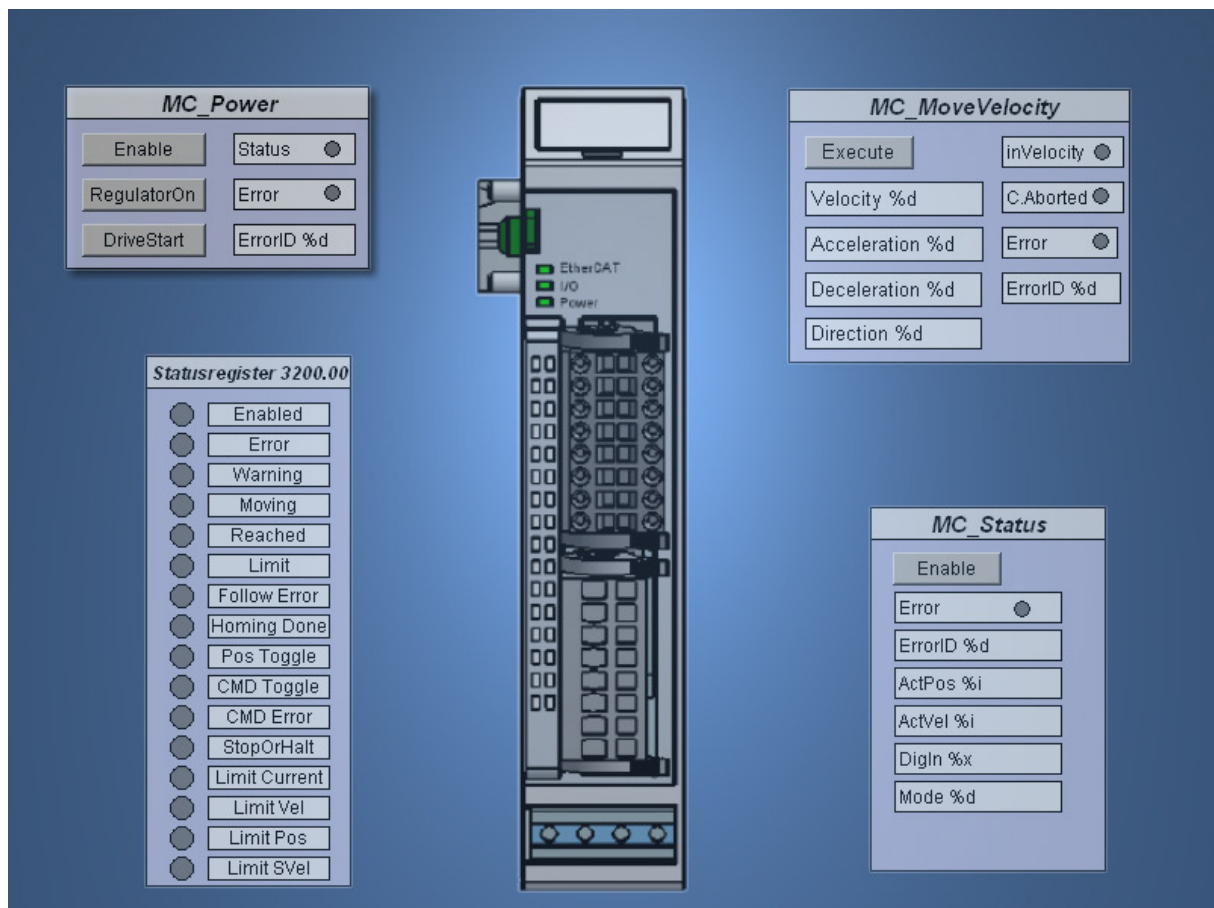


Kuhnke Elektronik Bedienhinweis

MC_xxx Codesys Bausteine für FIO Drive Control



1.0 Einleitung

Die Bibliothek LIB_FDC unterstützt das Kuhnke FIO Drive Control in der Programmierumgebung Codesys V2.3.x.x.

Alle relevanten Funktionen sind in diesen Bausteinen gekapselt, so dass auch ohne Kenntnisse von EtherCAT eine einfache Einbindung in die Software möglich ist.

Getestet und entwickelt wurden diese Bausteine für die Kuhnke INTime Systeme sowie für den Kuhnke Skaleo.

Exemplarisch wird hier die Installation unter Kuhnke INTime beschrieben.

2.0 Motion-Bausteine

MC_Power_FDC

Dieser Baustein kontrolliert die Leistungszuschaltung und den Bremsenzustand des Antriebs. Wurde ein Antrieb nicht auf diesem Weg eingeschaltet, keine Reglerfreigabe gesetzt oder die Bremse nicht gelöst, kann keine Bewegungssteuerung auf ihm erfolgen. Dieser Baustein muss für jede Achse existieren und ständig durchlaufen werden!

MC_ReadParameter_FDC

Mit diesem Baustein lässt sich ein Parameter aus der Antriebsstruktur lesen. Deren Parameternummern sind im Anhang beschrieben

MC_WriteParameter_FDC

Dieser Baustein schreibt einen Parameter in die Antriebsstruktur. Deren Parameternummern sind im Anhang beschrieben

MC_WriteListParameter_FDC

Dieser Baustein schreibt bis zu n Parametern in die Antriebsstruktur. Deren Parameternummern sind im Anhang beschrieben
Siehe auch : Beispielprogramm „Parameter übertragen“

MC_MoveAbsolute_FDC

Dieser Baustein bewegt die Achse an eine absolute Position entsprechend der vorgegebenen Werte von Geschwindigkeit, Bremsung und Beschleunigung. Bei einer Linearachse hat die Angabe der Richtung keine Bedeutung, bei Rotationsachsen bestimmt sie den Drehsinn.

MC_MoveRelative_FDC

Dieser Baustein bewegt die Achse um eine relative Distanz entsprechend der vorgegebenen Werte von Geschwindigkeit, Beschleunigung und Bremsung. Die Distanz kann dabei positiv oder negativ sein.

MC_MoveVelocity_FDC

Dieser Baustein bewirkt eine endlose Bewegung der Achse mit vorgegebener Geschwindigkeit. Um diese Geschwindigkeit zu erreichen, hält sich MC_MoveVelocity an die programmierten Werte von Beschleunigung und Bremsung. Die Sollgeschwindigkeit ist immer positiv. Die Eingangsvariable Direction legt die Richtung fest.

MC_Move_FDC

Dieser Baustein integriert die Bausteine MC_MoveAbsolute_FDC, MC_MoveRelative_FDC und MC_MoveVelocity_FDC. Bei einer hohen Anzahl von Achsen ist dieser Bausteinen zu empfehlen um Ressourcen zu schonen.

MC_MoveSync_FDC

Dieser Baustein verfährt die Achse im Interpolated Position Mode. Eine absolute Sollwertvorgabe wird vom Regler innerhalb einer fest eingestellten Zeit verfahren. Im Hauptprogramm muss eine Timertask eingerichtet werden. Siehe Beispiele Sync-Funktion.

MC_Home_FDC

Dieser Baustein stößt eine Referenzfahrt im Antrieb an. Alle Optionen der Referenzfahrt können angegeben werden.

MC_Reset_FDC

Dieser Baustein der setzt den Achs-Status von error_stop zurück auf standstill und muss nach jedem gemeldeten Fehler aufgerufen werden, damit wieder verfahren werden kann.

MC_Stop_FDC

Dieser Baustein bremst die Achse auf Geschwindigkeit 0. Dieser Baustein kann nicht unterbrochen werden und blockiert die Achse so lange wie der Eingang Execute gesetzt und die Achse noch nicht vollständig abgebremst ist.

MC_StartupDrive_FDC

Dieser Baustein beinhaltet alle Basisbausteine um eine Inbetriebnahme der Achse auszuführen.

2.1 Generelles zu den MC_ Bausteinen

Bausteine werden auf zweierlei Arten aktiviert:

a) **Enable- Eingang:** Verfügt der Baustein über einen Enable- Eingang (wie z.B. MC_ReadParameter_FDC), ist er genau solange aktiv, wie Enable TRUE ist.

b) **Execute- Eingang:** Der Baustein wird durch eine steigende Flanke (Übergang von FALSE nach TRUE) des Execute- Eingangs aktiviert und wird erst dann wieder inaktiv, falls er die Bewegung abgeschlossen hat, die Kontrolle über die Achse (AXIS_REF) kann ihm durch einen anderen Baustein entzogen wird, oder wenn er eine neue steigende Flanke am Execute- Eingang erhält und die Bewegung damit neu startet. Man beachte auch, dass alle Eingangs-Variablen nur bei steigender Flanke übernommen werden.

Die Bausteine zeigen über den Done- Ausgang (oder einen sinngemäßen Ausgang) entweder die Gültigkeit der Ausgänge (z.B. MC_ReadStatus) oder den Abschluss der Bewegung (z.B. MC_MoveAbsolute) an.

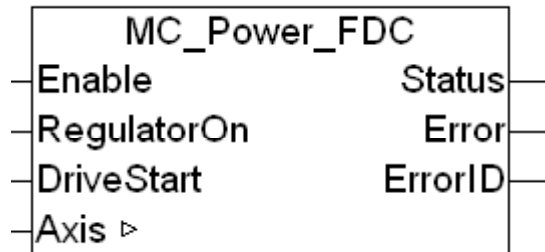
Ein bewegungserzeugender Baustein, der von einem anderen unterbrochen wird, setzt seinen CommandAborted- Ausgang, um dies anzuzeigen.

Die Ausgänge der Execute- gestarteten Bausteine bleiben – nach dem Setzen ihres Done- Ausgangs - so lange unverändert, wie der Execute- Eingang gesetzt wird. Durch eine fallende Flanke werden sie gelöscht. Wurde noch vor Beendigung eine fallende Flanke detektiert, so werden die Ausgänge für einen Zyklus gesetzt und im nächsten gelöscht.

Alle bewegungserzeugenden Bausteine verlangen dass in der entsprechenden Achse die Reglerfreigabe erteilt und die Bremse gelöst ist. Ansonsten melden Sie einen Fehler.

2.2 Die Bausteine im Detail

MC_Power_FDC



Enable : BOOL (Default: FALSE)
Solange diese Variable TRUE ist, ist der Antrieb eingeschaltet.

RegulatorOn : BOOL (Default: FALSE)
Schaltet die Lageregelung ein/aus.

DriveStart : BOOL (Default: FALSE)
Setzt bzw. löst die Bremse im Antrieb.

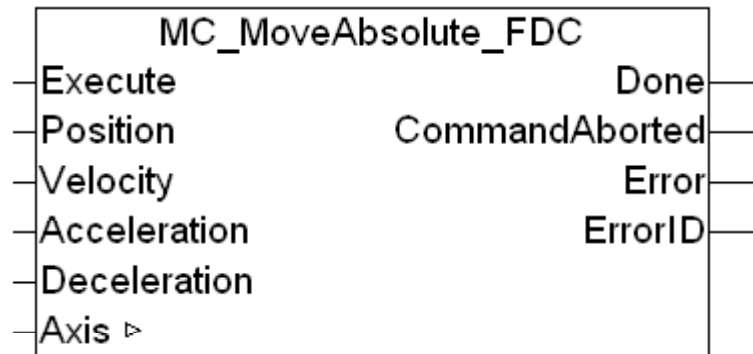
Axis : AXIS_REF_FDC
Hier wird die Struktur der Achse übergeben, die mit den Daten der Achse gefüllt wird.

Status : BOOL (Default: FALSE)
Zeigt an, ob der Antrieb augenblicklich in (TRUE) oder ohne Regelung (FALSE) ist.

Error : BOOL (Default: FALSE)
TRUE zeigt an, dass im Funktionsblock ein Fehler aufgetreten ist.

ErrorID : INT
Fehlernummer. Siehe Fehlerliste im Anhang.

MC_MoveAbsolute_FDC



Execute : BOOL

Bei einer steigenden Flanke wird der Baustein aktiv

Position : DINT

Absolute Sollposition für die Bewegung in Motoreinheiten je nach Parametrierung. Siehe Parameter \$6094 im Anhang. Änderungen während Fahrt möglich.

Velocity : INT

Geschwindigkeit für die Bewegung in U/Min. Änderungen während Fahrt möglich.

Acceleration : DINT (Default: 300)

Startrampe für die Bewegung in benutzerdefinierten Beschleunigungs Einheiten.

Deceleration : DINT (Default: 300)

Stopprampe für die Bewegung in benutzerdefinierten Beschleunigungs Einheiten.

Axis : AXIS_REF_FDC

Hier wird die Struktur der Achse übergeben, die mit den Daten der Achse gefüllt wird. Siehe auch Anhang Beschreibung Struktur AXIS_REF_FDC;

Done : BOOL

True wenn Achse Zielposition erreicht hat und in Lageregelung ist.

CommandAborted : BOOL

TRUE zeigt an, dass die Fahrt unterbrochen wurde. (MC_STOP_FDC)

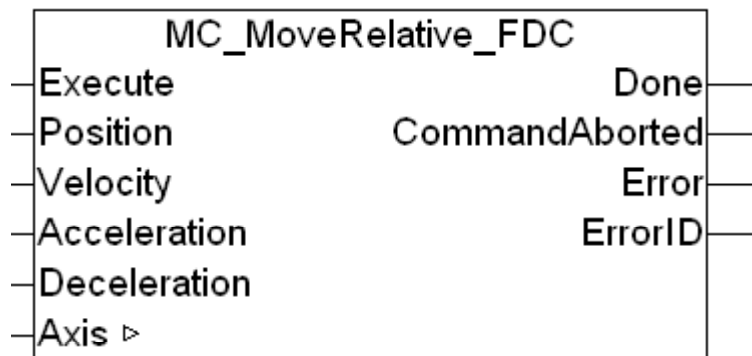
Error: BOOL (Default: FALSE)

TRUE zeigt an, dass im Funktionsblock ein Fehler aufgetreten ist.

ErrorID : INT

Fehlernummer. Siehe Fehlerliste im Anhang.

MC_MoveRelative_FDC



Execute: BOOL (Default: FALSE)

Bei einer steigenden Flanke wird der Baustein aktiv

Position: DINT

Relative Sollposition für die Bewegung in Motoreinheiten je nach Parametrierung. Siehe Parameter \$6094 im Anhang. Änderungen während Fahrt möglich.

Velocity: INT

Geschwindigkeit für die Bewegung in U/Min. Änderungen während Fahrt möglich.

Acceleration: DINT

Startrampe für die Bewegung in benutzerdefinierten Beschleunigungseinheiten.

Deceleration: DINT

Stopprampe für die Bewegung in benutzerdefinierten Beschleunigungseinheiten.

Axis: AXIS_REF_FDC

Hier wird die Struktur der Achse übergeben, die mit den Daten der Achse gefüllt wird. Siehe auch Anhang Beschreibung Struktur AXIS_REF_FDC;

Done: BOOL

True wenn Achse Zielposition erreicht hat und in Lageregelung ist.

CommandAborted: BOOL

TRUE zeigt an, dass die Fahrt unterbrochen wurde. (MC_STOP_FDC)

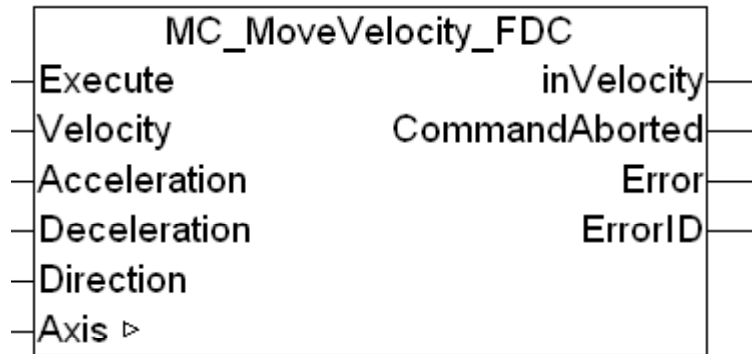
Error: BOOL (Default: FALSE)

TRUE zeigt an, dass im Funktionsblock ein Fehler aufgetreten ist.

ErrorID: INT

Fehlernummer. Siehe Fehlerliste im Anhang.

MC_MoveVelocity_FDC



Execute : BOOL (Default: FALSE)

Bei einer steigenden Flanke wird der Baustein aktiv und die Achse bewegt sich im Drehzahlmode.

Velocity : INT (Default: 1000)

Geschwindigkeit für die Bewegung in U/Min. Änderungen während Fahrt möglich.

Acceleration : DINT

Startrampe für die Bewegung in benutzerdefinierten Beschleunigungs Einheiten.

Deceleration : DINT

Stopprampe für die Bewegung in benutzerdefinierten Beschleunigungs Einheiten.

Direction : INT (Default: 1)

Richtung der Achse. -1=negativ +1=positiv. Änderungen während Fahrt möglich.

Axis : AXIS_REF_FDC

Hier wird die Struktur der Achse übergeben, die mit den Daten der Achse gefüllt wird. Siehe auch Anhang Beschreibung Struktur AXIS_REF_FDC;

InVelocity : BOOL

True wenn Achse Drehzahl erreicht hat.

CommandAborted : BOOL

TRUE zeigt an, dass die Fahrt unterbrochen wurde. (MC_STOP_FDC)

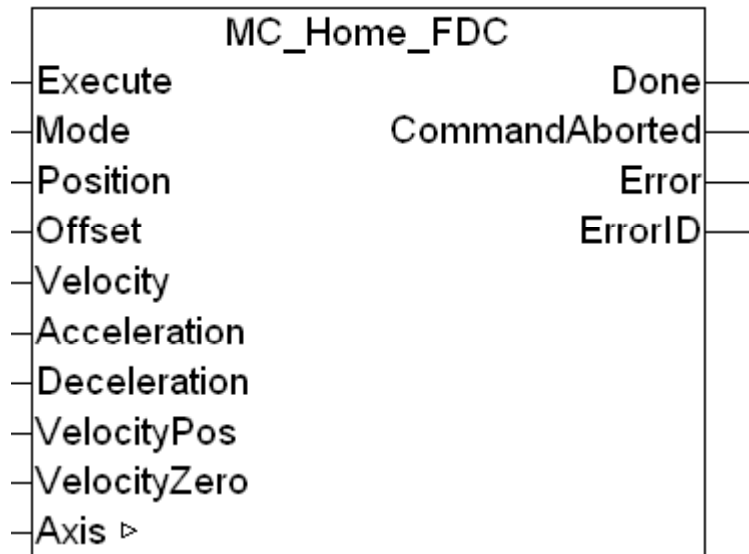
Error : BOOL (Default: FALSE)

TRUE zeigt an, dass im Funktionsblock ein Fehler aufgetreten ist.

ErrorID : INT

Fehlernummer. Siehe Fehlerliste im Anhang.

MC_Home_FDC



Execute : BOOL

Bei einer steigenden Flanke wird der Baustein aktiv

Mode : Byte

Refmode. siehe Anhang

Position : DINT

absolute Position auf die nach Erreichen des Referenzpunktes gefahren wird.

Offset : DINT

Istposition an der Referenzmarke

Velocity : INT

Geschwindigkeit für die Bewegung in U/Min.

Acceleration : DINT

Startrampe für die Bewegung in benutzerdefinierten Beschleunigungs Einheiten.

VelocityPos : INT

Geschwindigkeit für die Fahrt nach **Position**. 0 = keine Fahrt.

VelocityZero : INT ;

Geschwindigkeit für die Fahrt nach Nullposition.

Axis : AXIS_REF_FDC

Hier wird die Struktur der Achse übergeben, die mit den Daten der Achse gefüllt wird.
Siehe auch Anhang Beschreibung Struktur AXIS_REF_FDC;

Done : BOOL

True wenn Achse Zielposition erreicht hat und in Lageregelung ist.

CommandAborted : BOOL

TRUE zeigt an, dass die Fahrt unterbrochen wurde. (MC_STOP_FDC)

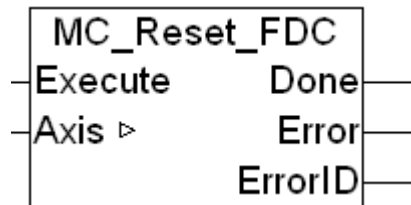
Error : BOOL (Default: FALSE)

TRUE zeigt an, dass im Funktionsblock ein Fehler aufgetreten ist.

ErrorID : INT

Fehlernummer. Siehe Fehlerliste im Anhang.

MC_Reset_FDC



Execute : BOOL (Default: FALSE)

Bei einer steigenden Flanke wird der Baustein aktiv und alle aktuellen anstehenden Fehler werden gelöscht. Solange ein Fehler ansteht kann keine Fahrbewegung durchgeführt werden !

Axis : AXIS_REF_FDC

Hier wird die Struktur der Achse übergeben, die mit den Daten der Achse gefüllt wird. Siehe auch Anhang Beschreibung Struktur AXIS_REF_FDC;

Done : BOOL

True wenn Fehler quittiert ist.

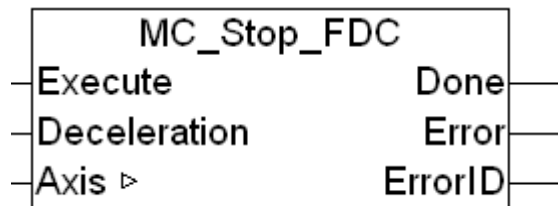
Error : BOOL (Default: FALSE)

TRUE zeigt an, dass im Funktionsblock ein Fehler aufgetreten ist.

ErrorID : INT

Fehlernummer. Siehe Fehlerliste im Anhang.

MC_Stop_FDC



Execute : BOOL (Default: FALSE)

Bei einer steigenden Flanke wird der Baustein aktiv und bricht die Fahrbewegung ab.

Deceleration : INT (Default: 1)

0 - Stop ohne Rampe. (Notstopp) MC_Power geht in Grundzustand

1- Stop mit Rampe.

Axis : AXIS_REF_FDC

Hier wird die Struktur der Achse übergeben, die mit den Daten der Achse gefüllt wird.

Siehe auch Anhang Beschreibung Struktur AXIS_REF_FDC;

Done : BOOL

True wenn Fehler quittiert ist.

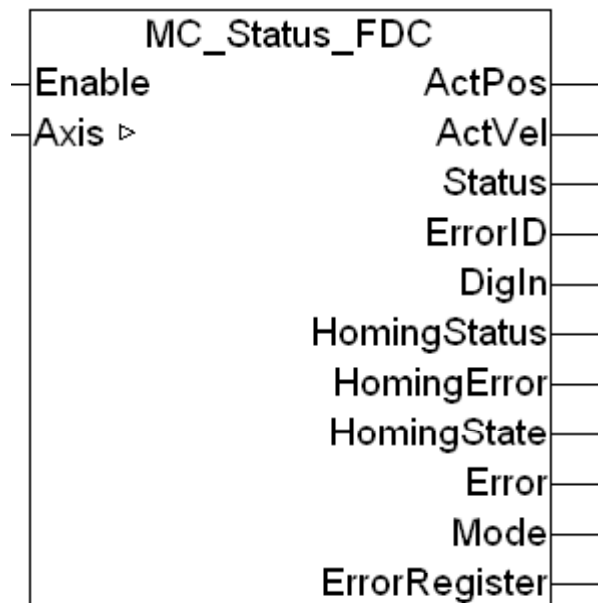
Error : BOOL (Default: FALSE)

TRUE zeigt an, dass im Funktionsblock ein Fehler aufgetreten ist.

ErrorID : INT

Fehlernummer. Siehe Fehlerliste im Anhang.

MC_Status_FDC



Die Werte können ebenso über Axis.<Name> abgerufen werden !

Enable : BOOL True – Werte werden gelesen.

Axis : AXIS_REF_FDC

Hier wird die Struktur der Achse übergeben, die mit den Daten der Achse gefüllt wird. Siehe auch Anhang Beschreibung Struktur AXIS_REF_FDC;

ActPos: DINT Aktuelle Ist-Position

ActVel: INT Aktuelle Ist-Drehzahl in [U./Min.]

Status: Word Statuswort des Reglers. Objekt 3002

ErrorID : INT Aktuelle Fehlernummer der Achse. Objekt 3001 .Siehe Fehlerliste im Anhang.

DigIn : Word Zustand der digitalen Eingänge. Objekt 3120

HomingStatus : Byte Status Referenzfahrt. Objekt 37B1,00

HomingError : Word Fehler Referenzfahrt. Objekt 37B1,01

HomingState : Byte Zustand Referenzfahrt. Objekt 37B1,02

Error : BOOL TRUE zeigt an, dass in der Fehler mit der Achse aufgetreten ist.

Mode : Byte aktueller Bewegungszustand der Achse. Siehe AxisRef_FDC

ErrorRegister : WORD Fehlerwort Motor. Siehe Anhang Fehlerwort des Motors.

3.0 Fehlerlisten

3.1 Fehlerliste MC_xxx Bausteine (Nr. = Error ID)

Baustein	Nr.	Beschreibung
Alle	2	Achse hat keine Reglerfreigabe. RegulatorOn = 0.
	3	Beim Aktivieren des Bausteins stand noch ein Mot.Fehler an
	4	Achse war beim Start noch in Bewegung
	5	Hardware Endschalter
	6	Beim Aktivieren des Bausteins stand die Achse im Stop. Bit 11 Status
	7	Fehler FDC. Fehlernummer in Axis.wErrorRegister
MC_Move_Absolute	70	Fehler Parameter : Sollgeschwindigkeit = 0
	72	Motorfehler während der Fahrt aufgetreten
MC_MoveRelative	80	Fehler Parameter : Sollgeschwindigkeit = 0
	82	Motorfehler während der Fahrt aufgetreten
MC_Stop	91	Falscher Wert für Deceleration
MC_Home	100	Fehler Parameter : Sollgeschwindigkeit = 0
	102	Motorfehler während der Fahrt aufgetreten
MC_MoveVelocity	110	Fehler Parameter : Sollgeschwindigkeit = 0
	112	Motorfehler während der Fahrt aufgetreten
	113	Ungültiger Wert für Direction
MC_Power	10	Timeout beim Warten auf Togglebit „Posimode setzen“
MC_Reset	20	Timeout beim Warten auf Togglebit „Stop aufheben“

3.3 Fehlerwort des Motors : <Achse>.wErrorRegister

Der Parameter wErrorregister ist identisch mit dem ErrorRegister Objekt 3001.00 des Reglers.

Die Dokumentation hierzu befindet sich im Dokument **DsaPar_de_BETA.chm**

5.0 Beispiele

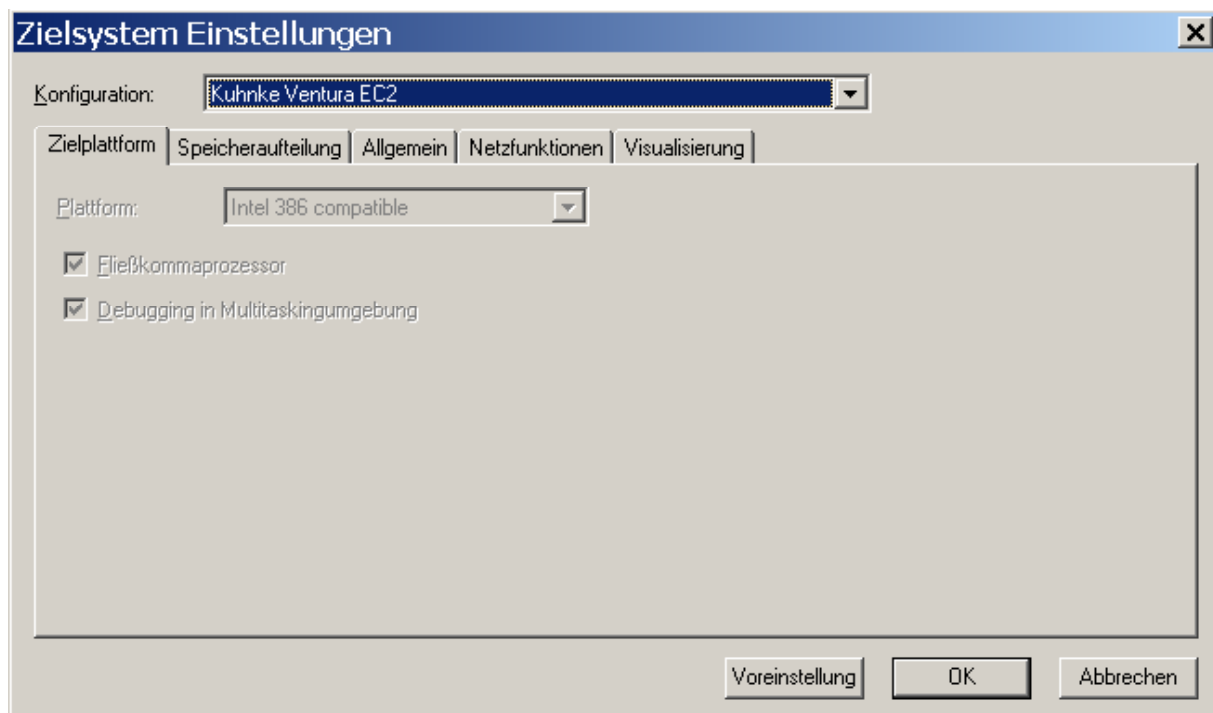
5.1 Beispielprogramm

Folgende Schritte sollten vorher erledigt sein.

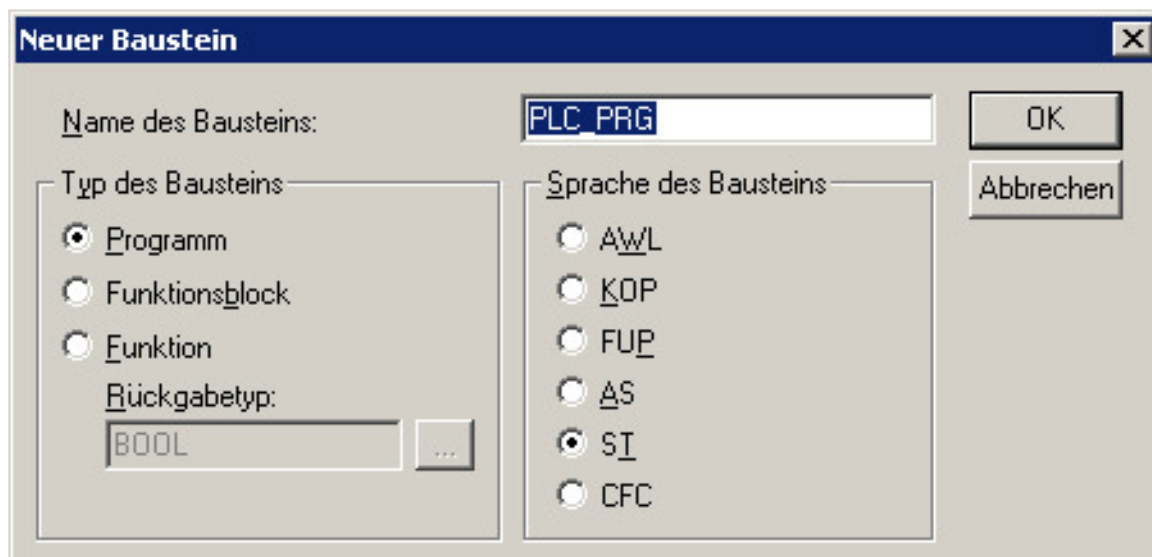
- Installation des jeweiligen Target für das gewünschte Zielsystem. Siehe 1.1
- Parametrieren der Achse mit dem McTools.
 - Achse im Menü **DSA Config** und **DSA Pid** konfigurieren und Parameter im Regler speichern.

Schritt 1 : Projekt anlegen.

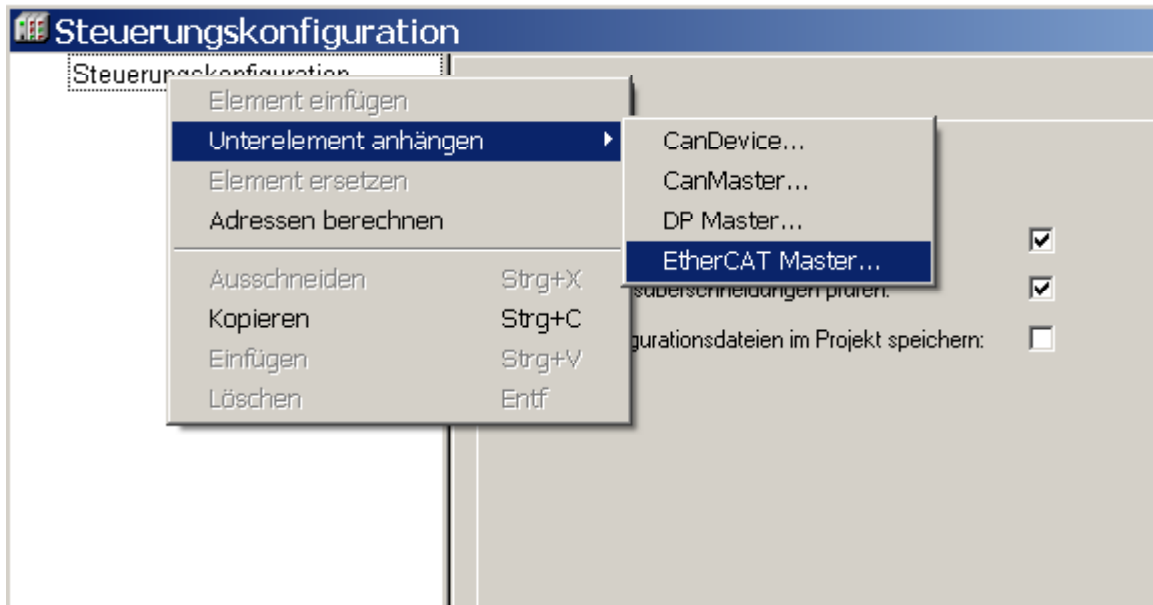
Erzeugen Sie ein neues Codesys Projekt. Wählen Sie als Zielsystem „Kuhnke Ventura EC2“.



Legen sie dann den Baustein PLC_PRG in ST an.



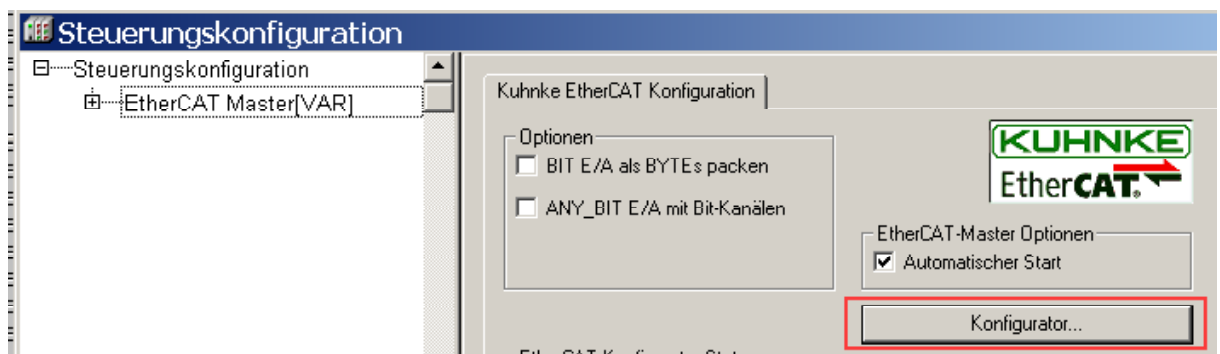
- Karte Resource wählen
- Klicken sie mit der rechten Maustaste auf Steuerungskonfiguration und wählen Sie Unterelement anhängen->EtherCAT Master



- Speichern sie das Projekt und klicken sie einmal auf Steuerungskonfiguration

Schritt 2 : EtherCAT konfigurieren

- Klicken sie in der Steuerungskonfiguration auf EtherCAT Master und dann rechts auf den Button „Konfigurator“



Der ETG Konfigurator öffnet sich, und das EtherCAT Bussystem kann nun konfiguriert werden.

Nach der Rückkehr aus dem Konfigurator werden automatisch die Objekte aller Teilnehmer in die Steuerungskonfiguration eingetragen

Schritt 3 : Bibliotheken einbinden

- Karte « Ressourcen » wählen
- Bibliotheksverwaltung wählen

Motion Bibliotheken einbinden

- rechter Mausklick in die Liste der Bibliotheken
- mit « weitere Bibliothek » LIB_FDC.lib hinzufügen

Schritt 4 : Achse und Motion-Control Bausteine definieren

Achse definieren

Für jede Achse muss ein Achsname vom Typ AXIS_REF_FDC sowie jeweils ein Zeiger auf den Anfang des Prozessabbildes (Input und Output) angelegt werden. Die Namen sind beliebig.

(* Deklaration Achse 1 *)

Fio1 : Axis_Ref_FDC;

gp_RX AT %QD2586 : TReceive_FDC; (* Startadresse Prozessabbild Output *)

gp_TX AT %ID2586 : TTransmit_FDC;(* Startadresse Prozessabbild Input *)

MC-Bausteine definieren

Legen sie nun von jedem Motionbaustein den Sie benötigen eine Instanz an. Für diesem Beispiel benötigen wir nur 3 Bausteine die wir im Baustein PLC_PRG anlegen.

MC_POWER : Basisbaustein zum Initialisieren der Achse. dieser Baustein muss immer vorhanden sein.

MC_MoveVelocity: Achse im Drehzahlmode verfahren

MC_STATUS: Überblick über den Zustand der Achse

```
0001 PROGRAM PLC_PRG
0002 VAR
0003     (* Instanzen der Motionbausteine *)
0004     Power :      MC_Power_FDC;
0005     Velocity :   MC_MoveVelocity_FDC;
0006     Status :    MC_Status_FDC;
0007
0008     Init: BOOL;
0009
0010 END_VAR
0011
```

Schritt 5 : Anwenderprogramm schreiben

Es soll ein kleines Programm geschrieben werden, welches einen Motor im Velocity-Mode fährt.

```
PROGRAM PLC_PRG
```

```
VAR
```

```
  (* Instanzen der Motionbausteine *)
```

```
  Power :      MC_Power_FDC;
```

```
  Velocity :   MC_MoveVelocity_FDC;
```

```
  Status :    MC_Status_FDC;
```

```
  Init: BOOL;
```

```
END_VAR
```

```
  (* Antriebsstruktur füllen *)
```

```
  IF NOT Init THEN
```

```
    Init := TRUE;
```

```
    Fio1.pRXpdo := ADR(gp_RX); (* Zeiger auf RX PDOs %Q Bereich *)
```

```
    Fio1.pTXpdo := ADR(gp_TX); (* Zeiger auf TX PDOs %I Bereich *)
```

```
  END_IF
```

```
  (* Aufruf der Motionbausteine *)
```

```
  Power(Axis := Fio1);
```

```
  Velocity(Axis := Fio1);
```

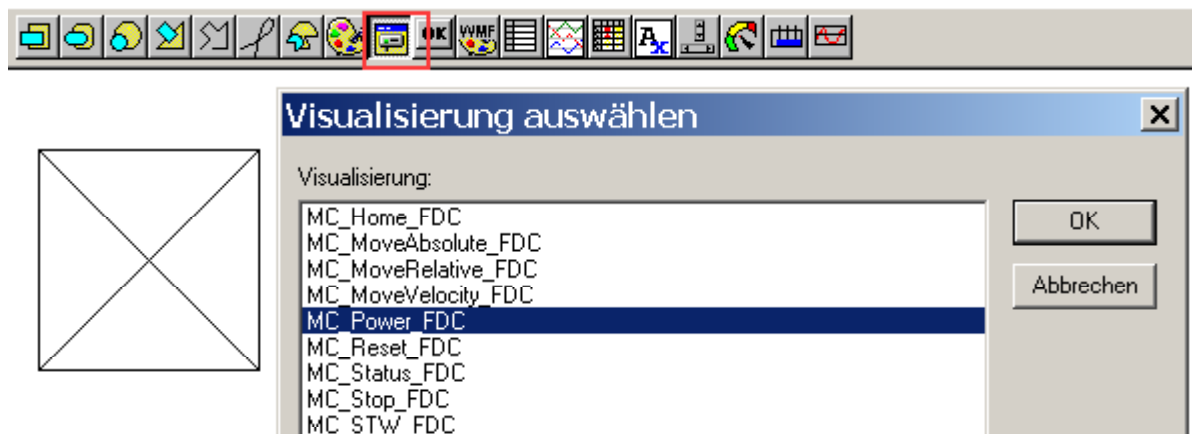
```
  Status(Axis := Fio1);
```

Schritt 6: Visualisierung erstellen

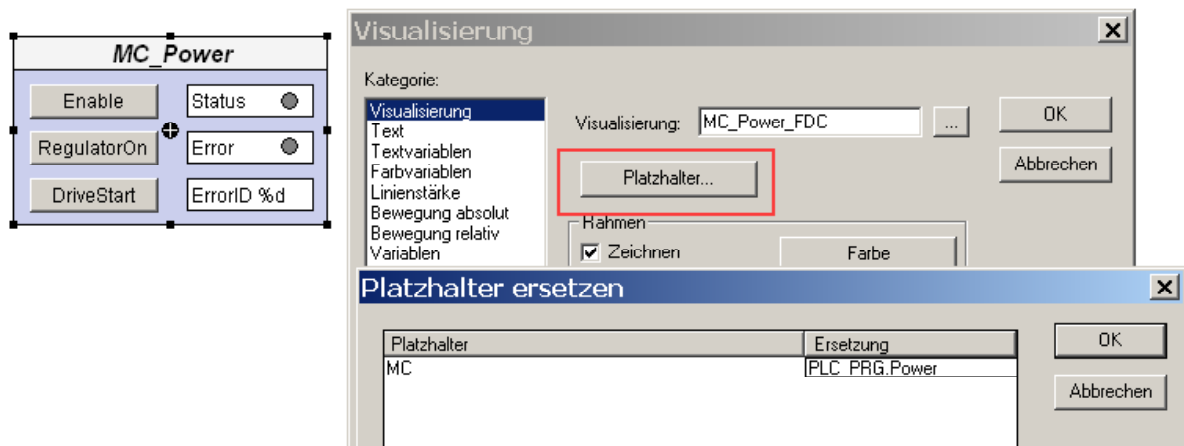
Zur Visualisierung existieren in der LIB_FDC für alle MC- Baustein Visualisierungstemplates. Darüber hinaus gibt es noch Templates für das Steuer- und Statuswort der Achse, sowie zum Senden von SDO's.

Visualisierungstemplate MC_Power

- Karte Visualisierung wählen und neues Objekt einfügen
- Visualisierungskomponente auf Zeichenfläche ziehen.
- Visualisierung „MC_Power_FDC wählen



- rechte Maustaste auf Komponente und Konfigurieren
- Klick auf Platzhalter -> Ersetzung für MC = PLC_PRG.Power



- Erstellen der Visualisierungstemplates

MC_MoveVelocity_FDC mit Platzhalter MC=PLC_PRG.Velocity
 MC_Status_FDC mit Platzhalter MC= PLC_PRG.Status

Schritt 7 : Online Programm übertragen und testen

- Menü Online -> Einloggen und Programm übertragen.
- Visualisierung wählen
- MC_Power Enable und RegulatorON einschalten
- MC_MoveVelocity Enable einschalten und die Achse dreht sich !

