



DOKUMENTATION ISG-kernel

Funktionsbeschreibung Jobmanager

Kurzbezeichnung:
FCT-M10

© Copyright
ISG Industrielle Steuerungstechnik GmbH
STEP, Gropiusplatz 10
D-70563 Stuttgart
Alle Rechte vorbehalten
www.isg-stuttgart.de
support@isg-stuttgart.de

Dokumentation Version: 1.01
05.06.2025

Allgemeine- und Sicherheitshinweise

Verwendete Symbole und ihre Bedeutung

In der vorliegenden Dokumentation werden die folgenden Symbole mit nebenstehendem Sicherheitshinweis und Text verwendet. Die (Sicherheits-) Hinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

Symbole im Erklärtext

- Gibt eine Aktion an.
- ⇒ Gibt eine Handlungsanweisung an.



⚠ GEFAHR

Akute Verletzungsgefahr!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht unmittelbare Gefahr für Leben und Gesundheit von Personen!



⚠ VORSICHT

Schädigung von Personen und Maschinen!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, können Personen und Maschinen geschädigt werden!



Achtung

Einschränkung oder Fehler

Dieses Symbol beschreibt Einschränkungen oder warnt vor Fehlern.



Hinweis

Tipps und weitere Hinweise

Dieses Symbol kennzeichnet Informationen, die zum grundsätzlichen Verständnis beitragen oder zusätzliche Hinweise geben.



Beispiel

Allgemeines Beispiel

Beispiel zu einem erklärten Sachverhalt.



Programmierbeispiel

NC-Programmierbeispiel

Programmierbeispiel (komplettes NC-Programm oder Programmsequenz) der beschriebenen Funktionalität bzw. des entsprechenden NC-Befehls.



Versionshinweis

Spezifischer Versionshinweis

Optionale, ggf. auch eingeschränkte Funktionalität. Die Verfügbarkeit dieser Funktionalität ist von der Konfiguration und dem Versionsumfang abhängig.

Inhaltsverzeichnis

Allgemeine- und Sicherheitshinweise	2
1 Übersicht	5
2 Beschreibung.....	6
2.1 Allgemein	6
2.2 Jobverwaltung	6
2.2.1 Jobstart	6
2.2.2 Jobstatus.....	8
2.2.3 Job-ID und Jobverfolgung.....	9
2.2.4 Jobabbruch	11
2.2.5 Jobende	13
2.2.6 Job-Stack	14
2.2.7 Priorisierung und Lock-Blöcke	15
2.2.8 Synchronisierung mit Auftragnehmer-Jobs.....	16
2.2.9 Jobmanager-Gruppen.....	16
2.3 Anwendungsszenarien des Jobmanagers	18
2.3.1 Mehrplatzbetrieb	18
2.3.2 Bearbeitung in mehreren Stationen	18
2.3.3 Senkerodieren.....	18
3 Programmierung.....	19
3.1 NC-Befehle des Jobmanagers	19
3.2 Kommandieren von Auftragnehmern	19
3.3 Sperren konkurrierender Auftraggeber	23
3.4 Warten auf Zustand von beauftragtem Job.....	24
3.5 Abfrage des zuletzt gültigen Status in #WAIT MC_Status	25
3.6 Kanalreset in einem beauftragten Kanal	26
3.7 V.G.Variablen.....	26
4 CNC-Objekte	28
4.1 MC_MovePath + MCV_GrpPathPrepare + MCV_GrpPathMode.....	28
4.2 MCV_GrpResetForce.....	29
4.3 MCV_GrpReadJobAck.....	29
5 PLCopen-Funktionsbausteine.....	30
5.1 MC_MovePath	30
5.2 MCV_GrpPathMode.....	32
5.3 MCV_GrpPathPrepare	34
5.4 MCV_GrpReadJobAck.....	37
5.5 MCV_GrpReadJobStatusList.....	39
5.6 MCV_GrpResetForced.....	40
5.7 MC_GrpStop	42
6 Parameter	44
7 Anhang	48
Stichwortverzeichnis.....	49

Abbildungsverzeichnis

Abb. 1:	Varianten, einen Job zu starten	7
Abb. 2:	Durchreichen der Job-ID aus dem Auftragsstart bis in die Anzeige einer Jobstatus-Änderung	10
Abb. 3:	Wirkung von Jobbrüchen auf Auftraggeber und Auftragnehmer	11
Abb. 4:	Jobende-Nachrichten aufgrund von M30 im Kontext von Auftraggeber und Auftragnehmer	13
Abb. 5:	Kanalzuordnung zu Jobmanager-Gruppen	17
Abb. 6:	Werte der V.G.Variable V.G.IP_NR im Kontext von auftraggebendem und auftragnehmenden Kanal	27

1 Übersicht

Aufgabe

Der Jobmanager (JM) koordiniert den Start von NC-Programmen in anderen Kanälen. Ein sogenannter Batchbetrieb wird möglich. Ein oder mehrere Kanäle sind mit ihren CNC-Programmen die Auftraggeber (früher: Master), andere Kanäle sind auftragnehmende Kanäle (früher: Slave).

Neben dem kontrollierten Weiterleiten von Aufträgen verwaltet der Jobmanager Rückmeldungen der Auftragnehmer. Wurden in einem Auftrag mehrere (unterlagerte) Auftragnehmer beauftragt, so werden deren Rückmeldungen zusammengefasst und dem Auftraggeber als Sammelmeldung mitgeteilt.

Der Jobmanager nutzt durch die PLCopen Organisation spezifizierte Stati und Basis-Funktionen und erweitert sie mit Funktionen zur Jobidentifikation und -verwaltung.



Versionshinweis

Die Funktionalität ist verfügbar ab CNC-Version V3.1.3110

Parametrierung

In der Hochlaufliste werden Kanäle als Auftraggeber oder Auftragnehmer konfiguriert.

Programmierung

Im Kapitel Programmierung [▶ 19] sind für auftraggebende Kanäle zusätzliche Befehle beschrieben.

Obligatorischer Hinweis zu Verweisen auf andere Dokumente

Zwecks Übersichtlichkeit wird eine verkürzte Darstellung der Verweise (Links) auf andere Dokumente bzw. Parameter gewählt, z.B. [PROG] für Programmieranleitung oder P-AXIS-00001 für einen Achsparameter.

Technisch bedingt funktionieren diese Verweise nur in der Online-Hilfe (HTML5, CHM), allerdings nicht in PDF-Dateien, da PDF keine dokumentenübergreifenden Verlinkungen unterstützt.

2 Beschreibung

2.1 Allgemein

Mehrkanalige CNC-Systeme werden häufig durch aufwendig programmierte PLC-Programme koordiniert. Diese PLC-Programme:

- starten parallelisierte oder streng sequentielle Vorgänge,
- kommandieren synchrone bzw. asynchrone Bewegungen in CNC Kanälen und
- führen Vorgänge und Bewegungen wieder zusammen.

Meist sind sie für eine Anwendung optimiert. Ein „Batchbetrieb“ oder der Betrieb mit „Rezepturen“, also eine Kommandierung mit abweichenden Abläufen oder Regeln über ein vom Anwender veränderbares Programm, ist nicht möglich. Ursachen sind in den Ablauf encodierte Abhängigkeiten von

- Steuerungseigenschaften
- Anwendungsprogrammen
- der Synchronisation der Programme
- dem zu fertigenden Produkt selbst.

Der Einsatz des Jobmanagers erlaubt einen Batchbetrieb ohne diese PLC-Unterstützung. Ein oder mehrere CNC Kanäle übernehmen mit ihren CNC-Programmen die Aufgabe eines Kommandierenden, auch „Auftraggeber“ (Master) genannt. Die anderen Kanäle sind Diensterbringer, sogenannte „Auftragnehmer“ (Slaves). Zum Einsatz kommen im auftraggebenden Kanal zusätzliche Befehle für

- den Start von NC-Programmen in Auftragnehmern
- das Blockieren anderer Auftraggeber bei ihrer Kommandierung und
- das Synchronisieren des Programmablaufs im auftraggebenden Kanal mit seinen Auftragnehmern.

Alle nicht als Auftraggeber deklarierten Kanäle zeigen beim Aufruf der zusätzlichen Befehle eine Fehlermeldung.

2.2 Jobverwaltung

Primäre Aufgabe des Jobmanagers ist die Verwaltung von Aufträgen – im Folgenden als „Jobs“ bezeichnet. Als Job wird eine Einheit folgender Attribute verstanden:

- Zu startendes CNC-Programm. Es wird identifiziert über den CNC-Programmdateinamen.
- Startmodus des CNC-Programms,
- optionale Parameter/Werte zur Übergabe an das kommandierte Programm,
- eine vom Anwender definierte „Job-ID“ und „Source-ID“.

Bei aktiviertem Jobmanager besitzt jeder Kanal eine Job-Warteschlange, in der zu startende Jobs abgelegt werden. Ist der Kanal frei – läuft also im Decoder des Kanals kein Programm -, dann wird der nächste in der Warteschlange liegende Job ohne Anwenderrückfrage ausgeführt. Die Startreihenfolge entspricht der Sequenz beim Einlagern von Jobs. Das im Job beschriebene Programm wird - analog zu dem bisher bekannten Programmstart (siehe MCM Zustandsautomat) geladen und gestartet, zusätzlich werden optionale Startmodi und Parameter an das Programm übergeben. Mithilfe der Job-ID bleibt der Job für den Jobmanager und den Anwender über die gesamte Laufzeit hinweg identifizier- und verfolgbar.

2.2.1 Jobstart

Es existieren drei Möglichkeiten, einen Job in der Job-Warteschlange eines Kanals abzulegen. Werden die Möglichkeiten parallel genutzt, so erfolgt ihre Abarbeitung aus den nachfolgend beschriebenen Schnittstellen in absteigender Priorisierung. Ist die Job-Warteschlange voll, so muss der Auftraggeber warten. Implizit warten dann auch nachrangige Auftraggeber bzw. deren Jobs. Die nachfolgende Abbildung zeigt die Startvarianten a) bis c):

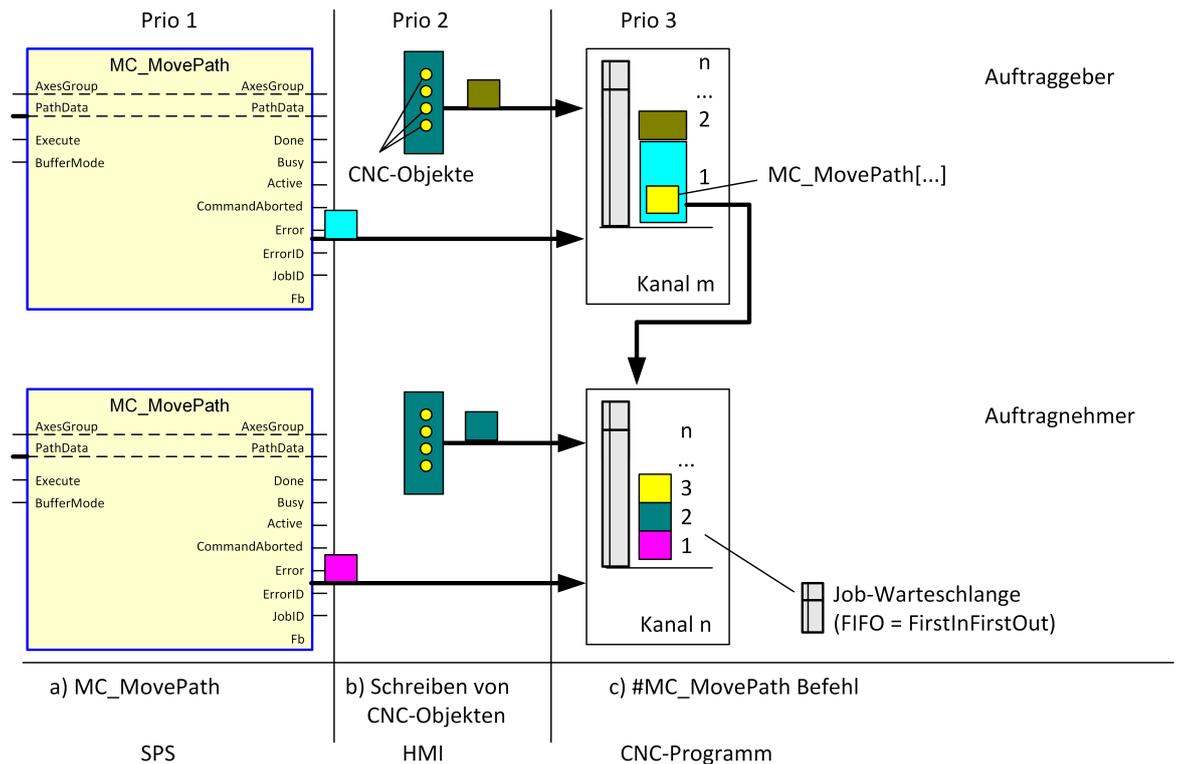


Abb. 1: Varianten, einen Job zu starten

- Variante a) Beauftragung über den (SPS) PLCopen Part4 Baustein MC_MovePath [▶ 30]. Der Baustein kann sowohl für auftraggebende als auch auftragnehmende Kanäle genutzt werden.
- Variante b) Beschreiben mehrerer CNC-Objekte in vorgegebener Reihenfolge. Ausgelöst wird die in a) beschriebene MC_MovePath Funktion. Typisch wird dieser Zugang von einer HMI genutzt. Die Schnittstelle kann für auftraggebende - und auftragnehmende Kanäle genutzt werden. (siehe CNC-Objekte [▶ 28])
- Variante c) Ein im auftraggebenden Kanal laufendes CNC Programm startet über die #MC_MovePath Anweisung einen Job in einem anderen auftragnehmenden Kanal. (siehe Kommandieren von Auftragnehmern [▶ 19])



Hinweis

Die Variante c) erfordert mindestens einen auftraggebenden und einen auftragnehmenden Kanal.

Variante a) ist nur über die SPS kommandierbar.

Die Varianten a) und b) sind unabhängig vom Jobmanagerbetrieb grundsätzlich in jedem CNC Kanal nutzbar.



Hinweis

Der Start eines NC-Programms über den bisher bekannten HLI MCM-Zustandsautomaten ist trotz des Jobmanagereinsatzes weiterhin möglich, aber nicht sinnvoll.

Unabhängig davon, ob im Auftragnehmer oder Auftraggeber gestartet, wird nur ein einzelnes Programm ohne Parameter und ohne Jobidentifikation ausgeführt. Der über den MCM Automat kommandierte Programmstart wird nicht in der Job-Warteschlange abgelegt. Er kennt keine Jobzustände sondern nur die bekannten und dokumentierten (Kanal-) Zustände.



Achtung

Bei Verwendung von beiden Kommandierungsarten ist der Anwender dafür verantwortlich, Deadlocks oder Kollisionen mit Jobmanager-Aufträgen zu vermeiden.

2.2.2

Jobstatus

Der Status jedes Jobs wird durch PLCopen definierte Zustände beschrieben, die über die gesamte Lebenszeit eines Jobs folgende Werte einnehmen können:

- „Busy“ (MC_BUSY): Der Job ist eingeplant, er liegt jedoch in der Warteschlange,
- „Started“ (MC_STARTED): Der Job wurde im Dekoder gestartet,
- „Aborted“ (MC_ABORTED): Der Job wurde abgebrochen. Voraus ging ein Stopp oder Reset durch den Anwender.
- „Error“ (MC_ERROR): Während einer beliebigen Phase der Jobbearbeitung wurde ein Fehler detektiert. Der Fehler kann im auftraggebenden Kanal selbst oder in einem von ihm gestarteten auftragnehmenden Kanal aufgetreten sein.
- „Done“ (MC_DONE): Fehlerfreie Beendigung des Jobs und seiner von ihm eventuell gestarteten auftragnehmenden Kanäle. Mit der Signalisierung dieses Status wird der Job aus der Verwaltung des Jobmanagers ausgetragen und gelöscht.



Hinweis

PLCopen verwendet zur Anzeige der Zustände eigenständige Ausgangsschnittstellen mit dem Wertebereich True/False. Die hier diskutierte Realisierung weicht teilweise davon ab und nutzt für die Zustände eine Enumeration / einen Zahlenwert. Der Name folgt der PLCopen Notation wobei der Eindeutigkeit halber „MC_“ (für Motion Control) vorangestellt ist.

2.2.3 Job-ID und Jobverfolgung

Die PLCopen Spezifikation trifft keine Vorgaben, wie die Fertigmeldung – also der Jobstatus - mehrerer, nacheinander über MC_MovePath kommandierter Aufträge an den Pins „Done“, „Busy“ oder „Active“ identifiziert werden kann. Sie schreibt nur vor, dass die Fertig-Nachricht am Pin „Done“ des MC_MovePath erst dann aktiv werden darf, wenn kein Auftrag mehr aktiv ist - somit das System „leer“ ist.

Bei maximaler Maschinenauslastung und dem „Dauerbetrieb“ des auftraggebenden Kanals wird der Status „Done“ nie aktiv werden. Um trotzdem für jeden der kommandierten Jobs und jeden der Statuswechsel dieser Jobs eine Rückmeldung erhalten zu können, kann vom Anwender eine Job-ID (Nummer) vergeben werden.

Drei Bausteine dienen der Beauftragung bzw. der Abfrage von Jobs:

- MCV_GrpPathMode [▶ 32]
- MCV_GrpPathPrepare [▶ 34]
- MCV_GrpReadJobAck [▶ 37]

Baustein MCV_GrpPathMode

Dieser Baustein ist optional und beeinflusst den „Mode“ (Betriebsart) des zu startenden Jobs (=CNC-Programms) (siehe auch Kanalbetriebsart).

Baustein MCV_GrpPathPrepare

Der Baustein MCV_GrpPathPrepare übernimmt diesen Mode und belegt weitere vom Anwender festlegbare Jobattribute wie z.B. die Job-ID, die über den Pin JobID vorgegeben werden kann. Alle Attribute werden über den „PathData“ Pin an den Baustein MC_MovePath zum Start des Jobs übergeben.

Baustein MCV_GrpReadJobAck

Der Baustein MCV_GrpReadJobAck wird vom Pin „Fb“ des Bausteins MC_MovePath versorgt. Er signalisiert mit steigender Flanke auf dem Ausgang „Valid“, dass ein neuer Status angezeigt wird. Angezeigt wird für jeden über MC_MovePath gestarteten Job jeder Jobstatus-Wechsel nach MC_BUSY, MC_STARTED, MC_ABORTED, MC_DONE und MC_ERROR. Gleichzeitig wird die jeweilige, vom Anwender beim Start des Jobs vergebene, Job-ID angezeigt.

Nachfolgende Abbildung zeigt beispielhaft einen Job mit der Job-ID „1“, der ursprünglich über MCV_GrpPathPrepare gestartet wurde und gerade fertig wurde.

Im Baustein MCV_GrpReadJobAck stand zuletzt der Status MC_DONE dieses Jobs mit der Job-ID „1“ an.

Zuvor wurden über diesen Baustein direkt nach dem Kommandieren in MC_MovePath zuerst der Status MC_BUSY signalisiert, nach dem Start im Dekoder dann der Status MC_STARTED.

Nach dem Start des Jobs mit Job-ID „2“ lässt sich in MCV_GrpReadJobAck der Status MC_BUSY für die Job-ID „2“ auslesen. Später werden die Stati MC_STARTED, MC_DONE oder im Fehlerfall MC_ABORTED bzw. MC_ERROR folgen.

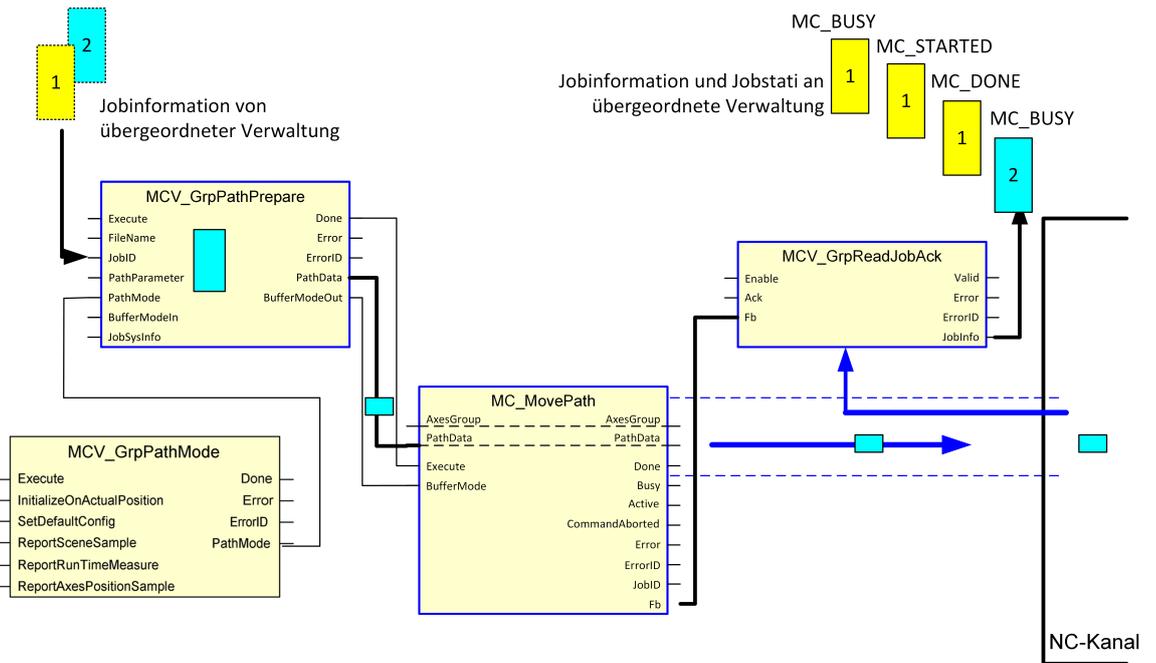


Abb. 2: Durchreichen der Job-ID aus dem Auftragsstart bis in die Anzeige einer Jobstatus-Änderung



Hinweis

Wird **MC_MovePath** in Verbindung mit **MCV_GrpPathPrepare** ausgeführt, muss permanent **MCV_GrpReadJobAck** für das Auslesen der Jobstati aufgerufen werden.

Erfolgt dies nicht, laufen interne Puffer voll und blockieren neu kommandierte Jobs.

Alternativ zu den SPS Bausteinen **MCV_GrpPathPrepare**, **MC_MovePath** und **MCV_GrpReadJobAck** existieren CNC-Objekte, die es erlauben, Jobs zu aktivieren und Statusmeldungen auszulesen. Siehe CNC-Objekte [28]



Hinweis

Wird die **MC_MovePath** Funktion über CNC-Objekte ausgelöst so müssen Quittungen über die entsprechenden CNC-Objekte des **MCV_GrpReadJobAck** zurückgelesen werden.

Erfolgt dies nicht, laufen Jobmanager-interne Puffer voll und blockieren neu kommandierte Jobs.



Hinweis

Wird **MC_MovePath** aus einem CNC Programm heraus gestartet, so kann der Status des beauftragten Jobs wie in Synchronisierung mit Auftragnehmer-Jobs [16] überwacht werden. Benötigt wird zum Aufruf seine Job-ID.



Hinweis

Für MCV_GrpResetForced existiert wie für MC_MovePath und MCV_GrpReadJob-Ack ein CNC-Objekt.

#MCV_GroupResetForced kann auch aus einem auftraggebenden CNC-Programm gestartet werden. Wie bei MC_MovePath ist vom Anwender eine Job-ID zu vergeben, um über die beschriebene Synchronisierungsfunktion [▶ 16] überwacht werden zu können. Das Abbruchverhalten entspricht dem durch die SPS ausgelösten #MCV_GroupResetForced auf einem auftragnehmenden Kanal.

2.2.5

Jobende

In der „klassischen Sicht“ eines CNC Programms terminiert M30 das laufende Programm und „stoppt“ den CNC-Kanal.

Bei der Nutzung des Jobmanagers beendet sich im *auftragnehmenden Kanal* bei M30 nicht nur das CNC-Programm, sondern wird - falls vorhanden – zusätzlich an den auftraggebenden Job bzw. an den Auftraggeber ein MC_DONE gesendet. Dieser nutzt die Nachricht zur internen Verwaltung seiner Jobs. Befindet sich ein weiterer Job in der Warteschlange des Auftragnehmers, wird dieser automatisch aktiviert (Nachricht MC_ACTIVATED an den Auftraggeber) und das im Job beschriebene Programm gestartet.

Terminiert im *auftraggebenden Kanal* hingegen ein Programm mit M30, startet dieser das Programm des nächsten wartenden Jobs. Der zum M30 gehörende Job wird jedoch erst dann beendet, wenn alle durch ihn, bzw. sein zuvor abgearbeitetes Programm, kommandierten Auftragnehmer-Jobs ebenfalls beendet sind. Ist dies der Fall, erfolgt ein MC_DONE an den Auftraggeber, also die auf dem Baustein MCV_GrpReadJobAck wartenden SPS. Weil anzunehmen ist, dass kommandierte Auftragnehmer-Jobs beim Start des nachfolgenden Jobs noch nicht beendet sind, sind in Folge mehrere Jobs aktiv.

Werden in einem Auftraggeber hintereinander Programme mit unterschiedlichen Auftragnehmern kommandiert, so ist es möglich, dass diese in der Reihenfolge „später“ gestartete Jobs zeitlich „vor“ den zuvor kommandierten Jobs fertig werden.

Nachfolgende Abbildung beschreibt den Zusammenhang von M30 und MC_DONE anhand eines Jobs „J1“ und dessen Unteraufträge „J1.1“ und „J1.2“. Beenden sich „J1.1“ und „J1.2“ mit M30, so schicken sie dem Auftraggeber ein MC_DONE. Wenn das Programm von „J1“ selbst mit M30 beendet wurde und alle beauftragten Jobs ebenfalls den Status MC_DONE haben, kann für „J1“ ein MC_DONE abgesetzt werden. Alle anderen Jobs und ihre zugehörigen Programme sind noch nicht beendet, da die entsprechenden MC_DONE fehlen.

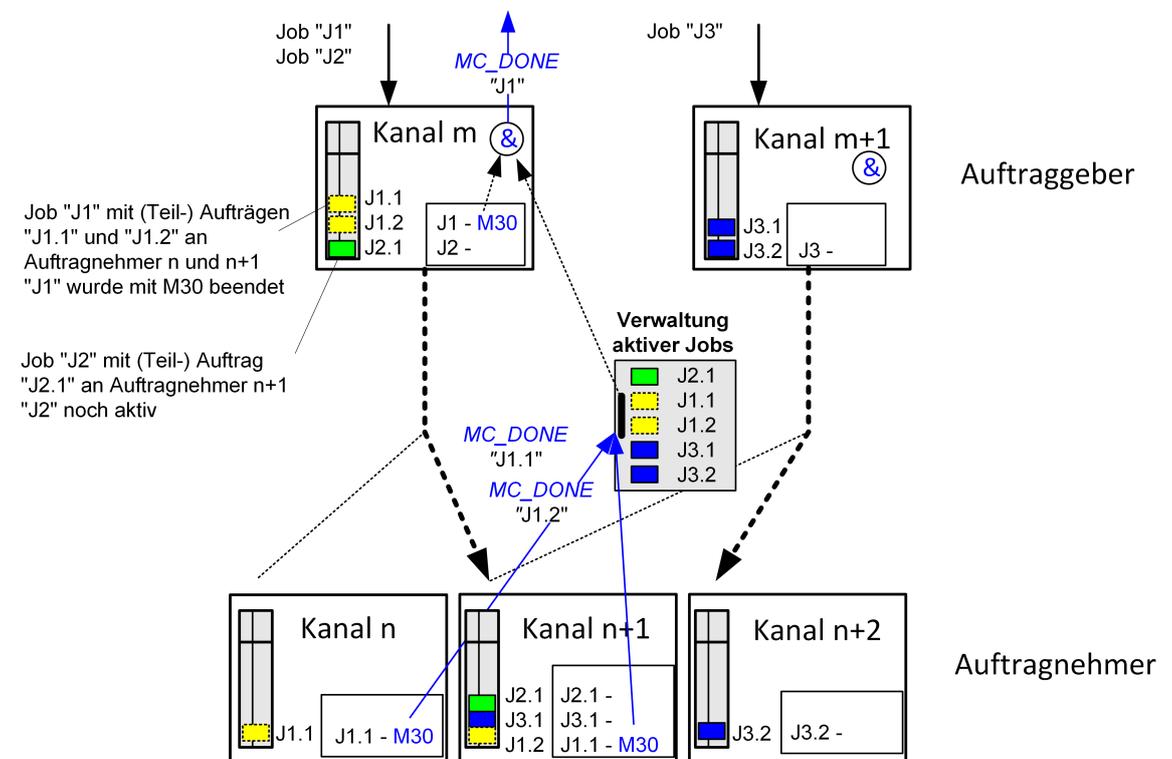


Abb. 4: Jobende-Nachrichten aufgrund von M30 im Kontext von Auftraggeber und Auftragnehmer

2.2.6

Job-Stack

Wie in Kapitel Job-ID und Jobverfolgung [► 9] beschrieben wird die vom Anwender bei Start eines Jobs vergebene Job-ID bei der Beendigung des Jobs zurückgeliefert. Die Job-ID sollte aber auch in allen vom Auftraggeber kommandierten Auftragnehmer-Jobs sichtbar werden, um zu Testzwecken oder auch zu Protokollierungszwecken verwendet werden zu können.

Der JM nutzt hierfür einen „Job-Stack“: Beim Anlegen eines neuen Jobs wird dieser erzeugt und über ein erstes „PUSH“ mit der Job-ID des Anwenders initialisiert. An jeder – auch steuerungsinternen - Kommandierungsebene wird der Stack mit einem weiteren „PUSH“ der jeweiligen internen Job-ID erweitert. Nach der Beendigung eines Jobs sendet der beauftragte Teilnehmer seinem Auftraggeber – zusammen mit dem Status MC_DONE, MC_ABORTED oder MC_ERROR – den aktuellen Job-Stack zurück. Der Empfänger der Nachricht erschließt sich seine Job-ID durch ein „POP“ und aktualisiert bei Bedarf seine Jobverwaltung.

Der Jobmanager unterstützt einen Multi-Auftraggeberbetrieb. Dies bedeutet, dass Auftragnehmer von mehreren Auftraggebern beauftragt werden können. Da der Job-Stack und der Jobstatus immer an den korrekten Auftraggeber zurückgesendet werden müssen, ist in jedem Eintrag im Job-Stack neben der Job-ID eine zusätzliche „Source-ID“ eingetragen. Diese Source-ID wird steuerungsinternen zur Routenfindung der Jobstati (Quittungen) genutzt. Sie wird auch im Baustein MCV_GrpReadJobAck angezeigt.

Voraussetzung für diese vom Anwender in MCV_GrpReadJobAck auslesbare „Source-ID“ ist die Vergabe eines Werts im MCV_GrpPathPrepare Baustein. Hier kann neben zwei durch den Anwender vergebaren Job-IDs eine eigene Source-ID (Nummer) belegt werden. MCV_GrpReadJobAck liefert bei jedem Statuswechsel alle drei Einträge zurück. Sowohl Source-ID als auch die beiden Job-IDs lassen sich, genauso wie die steuerungsintern vergebenen Job-IDs und Source-ID Tupel, im Job-Stack in allen Ebenen der Jobverarbeitung über Statusanzeigen bzw. im Trace verfolgen.



Hinweis

Die Job-ID kann vom Anwender zur Identifikation von Werkstücken in eigenen Produktionsprozessen genutzt werden. Die Source-ID unterstützt ihn dabei, bei alternativen Kommandogebern, Quittungen an diese direkt zurückadressieren zu können.

2.2.7 Priorisierung und Lock-Blöcke

Existieren mehrere auftraggebende Kanäle, die ein oder mehrere gemeinsam genutzte auftragnehmende Kanäle kommandieren, so konkurrieren deren Jobs - genauer deren Programme - darin, Aufträge an den gemeinsamen Auftragnehmer abzusetzen. Der Jobmanager bestimmt die Reihenfolge der Kommandierung:

1. Priorisiert wird der am längsten auf Zuteilung wartende Job. Genutzt wird hierzu ein steuerungsinerner Zeitstempel, der beim erstmaligen Zustellversuch jedes neuen Kommandos neu vergeben wird. Bei Zeitgleichheit wird der Kanal bevorzugt, der die kleinste Konfigurationsnummer „j“ im Parameter master[j].log_id von P-STUP-00206 enthält.
2. Kann der gewählte Auftraggeber trotz Zuteilung sein Kommando nicht ausführen (z.B. einen Job in die Auftragnehmer-Warteschlange legen - weil diese z.B. voll ist), so muss er warten. Alle anderen Auftraggeber warten ebenfalls. Es findet keine Neupriorisierung statt.
3. Das Warten kann nur durch einen Abbruch (MCV_GrpResetForced, MC_GrpStop) unterbrochen werden, welcher bevorzugt ausgeführt wird.

Ohne zusätzliche Eingriffe entsteht bei mehreren aktiven Auftraggebern – also mehreren gleichzeitig laufenden CNC Programmen - im Extremfall eine chaotische Abarbeitung konkurrierender Jobs.

Wenn dies nicht zulässig ist und ein auftraggebender Job in „seinem Batch“ exklusiven Zugriff auf die Reihenfolge der Kommandierung benötigt, so kann von ihm ein „Lock-Block“ erstellt werden. Die NC-Befehle hierfür sind:

- #LOCK und
- #UNLOCK

Mit Ausführung des #LOCK-Befehls blockiert ein Auftraggeber bis zum #UNLOCK - oder bis zu seinem Programmende mit M30 - alle anderen Auftraggeber dauerhaft in der Kommandierung ihrer Befehle. Die zuvor beschriebene Priorisierung ist solange deaktiviert. Möglich werden dadurch vorgegebene Abfolgen von Jobs in den Warteschlangen auftragnehmender Kanäle und Jobs in kooperierenden Kanälen, deren Jobs/Programme quasi „gleichzeitig“ und ohne Deadlock in den Auftragnehmern abgelegt werden können.



Hinweis

Der Lock-Block wirkt nur auf Jobmanager-spezifische Kommandos. Alle anderen CNC Kommandos werden unabhängig davon weiterhin ausgeführt.



Hinweis

Ist kein Lock-Block aktiv, so konkurriert ein neu kommandierter #LOCK Befehl mit allen gleichzeitigen Befehlen in anderen auftraggebenden Kanälen. Es gilt die zuvor beschriebene Priorisierungsregel.



Hinweis

Ein Reset-/ Stopp-Befehl durch einen beliebigen Auftraggeber wird unabhängig zu einem aktiven Lock-Block immer durchgesetzt. Ein aktiver Lock-Block wird dabei deaktiviert.

2.2.8 Synchronisierung mit Auftragnehmer-Jobs

Ein Auftraggeber-Job kann sich mit einem oder mehreren seiner beauftragten Jobs synchronisieren. Voraussetzung ist die vom Anwender beim Start über #MC_MovePath vergebene „Job-ID“.

Mit

```
#WAIT MC_Status [JobID=.. , ...]
```

und der bekannten Job-ID wird auf ein oder mehrere Ereignisse gewartet. Ereignisse sind bei #WAIT MC_Status ein oder mehrere der vier Stati MC_STARTED, MC_ABORTED, MC_ERROR, MC_DONE. Ist beim Aufruf keiner der Parameter angegeben, so entspricht dies der impliziten Vorgabe von MC_DONE und MC_ABORTED.



Hinweis

Job-IDs werden in einer Historie gespeichert, um auch nach Beendigung/Abbruch des Jobs bei einer „späteren“ Abfrage einen korrekten Status zurückliefern zu können. Ist die Job-ID unbekannt, da nie kommandiert oder nicht mehr in der Historie, so wird ein Fehler ausgegeben.



Hinweis

Die Job-ID Historie wird mit dem Beenden des Programms gelöscht.

Es kann daher nicht mehr auf Jobs vorhergegangener (Auftraggeber-) Jobs synchronisiert werden. Es gibt auch keine Möglichkeit, sich mit Jobs eines anderen Auftraggebers zu synchronisieren.

2.2.9 Jobmanager-Gruppen

Aktuell sind maximal zwei unabhängige Jobmanager mit ihren jeweiligen Auftraggeber- und Auftragnehmer-Kanälen konfigurierbar, wobei ein oder mehrere Kanäle als auftraggebender oder auftragnehmender Kanal festgelegt werden können. Ein Kanal kann nur einer der beiden Jobmanagergruppen angehören. In Folge kann ein auftragnehmender Kanal einer Jobmanagergruppe nicht gleichzeitig Auftraggeber einer anderen Jobmanagergruppe sein.



Hinweis

Ein Wechsel der Eigenschaft Auftragnehmer oder Auftraggeber oder der Wechsel von einer in die andere Jobmanagergruppe ist während des Steuerungsbetriebs nicht möglich.

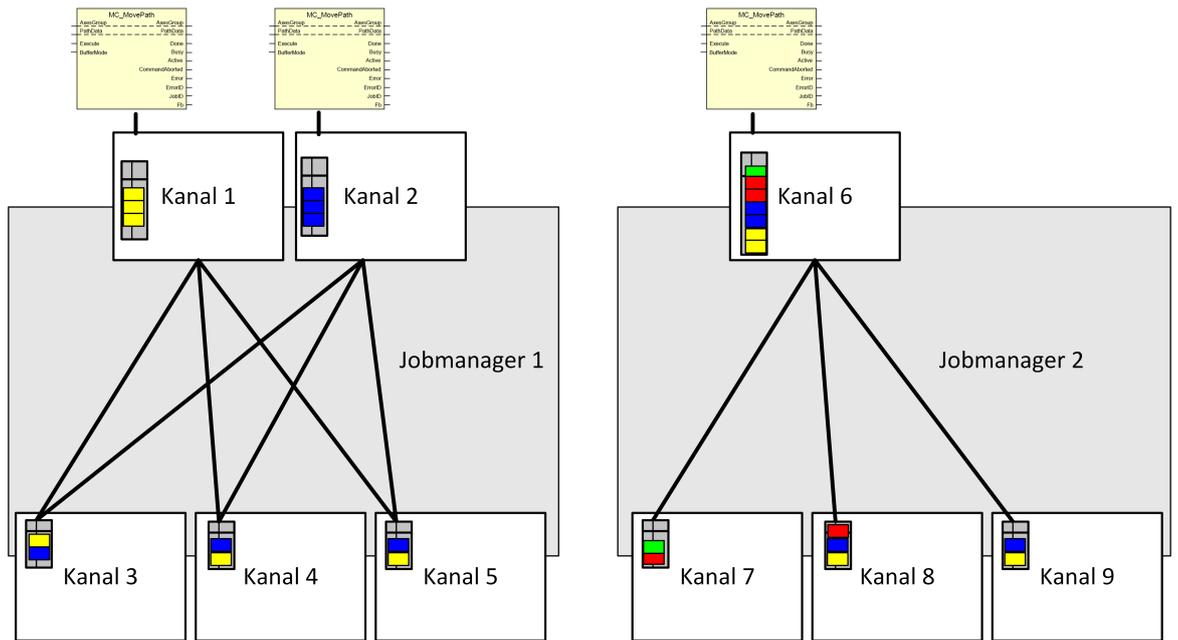


Abb. 5: Kanalzuordnung zu Jobmanager-Gruppen

2.3 Anwendungsszenarien des Jobmanagers

Die folgenden Beispiele nutzen den Jobmanager in abweichenden Anwendungsformen. Hierbei führt ein auftraggebender Kanal mit seinem Programm die „Rezeptur“ aus, jeder auftragnehmende Kanal erbringt vordefinierte Leistungen/Funktionen. Jeder der beteiligten Kanäle kann zeitgleich als vollwertiger CNC Kanal genutzt werden.

2.3.1 Mehrplatzbetrieb

Eine Maschine besitzt zwei Bearbeitungstische und ein Bearbeitungsportal. Das Portal arbeitet abwechselnd oder parallel an einem oder beiden Tischen befestigten Werkstück/Werkstücken. Die „Tische“ sind in dieser Konfiguration auftraggebende Kanäle, wobei jeder „Tisch“ seine Bearbeitungsfolge als Einzelaufträge an den Auftragnehmer, das Portal, kommandiert. Beide Tische lassen sich unabhängig voneinander rüsten und für die Fertigung freigeben. Zusammenhängende Aufträge werden je Tisch über einen Lock-Block exklusiv kommandiert.

Werden z.B. auch Werkzeugverwaltung oder Werkzeugwechsler als eigenständige Kanäle genutzt, so kann der auftraggebende Tisch über einen temporär nutzbaren Lock-Block untereinander kooperativ arbeitende Auftragnehmer „gleichzeitig“ kommandieren. So können z.B. „voraussehlend“ Werkzeuge in der Einwechselposition bereitgestellt werden, auf die das „parallel kommandierte“ CNC Programm des Bearbeitungsportals später zugreift. Die Neben- oder Rüstzeiten können somit reduziert werden.

Die in den Auftraggeberkanälen kommandierten Jobs entsprechen einem „Palettenauftrag“.

2.3.2 Bearbeitung in mehreren Stationen

Ein Auftraggeber-Job startet eine Folge von Arbeitsschritten (=Jobs) an einem Werkstück. Der Jobmanager verteilt diese Folge an weitgehend „autonome“, in ihrer Funktion unabhängige Bearbeitungseinheiten. Beispielfür eine Rundtaktmaschine wäre dies die Folge:

1. Transport_zu_1
2. Bearbeitung_1
3. Transport_1-2
4. Bearbeitung_2
5. Transport_2-x
6. ...

Da die Jobs in den Job-Warteschlangen der auftragnehmenden Kanäle abgelegt werden, ist der Auftraggeber nach seiner Kommandierung frei. Er kann weitere Nachfolge-Werkstücke (auch mit abweichender Bearbeitungsfolge) solange kommandieren, bis die Job-Warteschlange eines von ihm benötigten Auftragnehmers den Status „voll“ meldet. Er bleibt stehen. Wird der Platz frei, so setzt er seine Kommandierung fort. Mehrere Werkstücke (=Jobs) sind innerhalb einer Maschine „gleichzeitig“ in Bearbeitung.

Der im auftraggebenden Kanal kommandierte Job entspricht einem „Werkstückauftrag“.

2.3.3 Senkerodieren

Ein Sonderfall der Bearbeitung in mehreren Stationen stellt das „Senkerodieren“ (Siehe FCT-C44] dar. Hier nutzt ein auftraggebender Kanal, der Senkkanal, den Jobmanager, um zwei Auftragnehmerkanäle, den Planetärkanal und Rückzugskanal koordiniert mit NC-Programmen zu versorgen.

Besonderheit dieser Anwendung ist, dass durch die Überlagerung der jeweils zweidimensionalen Bewegungen des Senkkanals und Planetärkanal eine resultierende dreidimensionale Bewegung entsteht. Der Rückzugskanal überlagert zusätzlich die Bewegungen des Senkkanals.

Der im auftraggebenden Kanal kommandierte Job entspricht einem „Werkstückauftrag“.

3 Programmierung

3.1 NC-Befehle des Jobmanagers

Die nachfolgend beschriebenen Jobmanager-Befehle und -Konstanten können nur in Kanälen ausgeführt werden, die als Auftraggeber konfiguriert sind.

Werden diese Befehle in Kanälen ausgeführt, die als Auftragnehmer konfiguriert sind, dann wird der Fehler ID 280611 ausgegeben.

Bei allen Befehlen wird solange angehalten, bis der Auftrag erfolgreich abgeschlossen wurde: Bei #MC_MovePath und #MCV_GrpResetForced ist es das erfolgreiche Ablegen des Jobs in der Warteschlange, bei Synchronisierungsbefehlen der Eintritt des erwarteten Status.



Hinweis

Alle Befehle bzw. Parameternamen sind case-sensitiv!

3.2 Kommandieren von Auftragnehmern

Ein im auftraggebenden Kanal laufendes CNC Programm startet über die #MC_MovePath Anweisung einen Job in einem anderen „auftragnehmenden“ Kanal.



Versionshinweis

Verfügbar ab V3.1.3110

Syntax NC-Befehl:

```
#MC_MovePath [SYN] [ CH=.. JobID=.. FileName=.. @PL<1...20>=..
    [InitializeOnActualPosition=..]
    [SetDefaultConfig=..]
    [ReportSceneSampling=..]
    [ReportRunTimeMeasure=..]
    [ReportAxesPositionSample=.. ]
```

SYN	Synchrone Befehlsausführung des ISO Programms. Vor Ausführung des Befehls erfolgt eine implizite Kanalsynchronisation (implizites #FLUSH WAIT).						
CH=..	Logische Kanalnummer des Kanals, in dem der Auftrag ausgeführt werden soll. Die Verknüpfung zwischen CNC-Kanalnummer und den CNC-Kanälen wird in der Hochlaufliste angegeben. (Siehe Parameter [▶ 45]).						
JobID=..	Nutzerspezifische Auftrags-Identifikationsnummer (Job-ID). Die Jede Nummer muss eindeutig innerhalb des auftraggebenden Programms des Masterkanals sein. Diese ID wird z.B. im #WAIT MC_Status Befehl für die Auftragsidentifizierung verwendet.						
FileName=<Dateiname>	Dateiname des ISO Programms, welches gestartet werden soll.						
@P<i>=..	Parameterübergabe an das beauftragte ISO Programm. Auf die Parameter kann im aufgerufenen "Hauptprogramm" per @PL<i> zugegriffen werden.						
InitializeOnActualPosition=..	<p>Anfordern der aktuellen Position bei Programmstart. Siehe Kanalbetriebsart- SUPPRESS_POSITION_REQUEST</p> <p>Wird InitializeOnActualPosition nicht genutzt, so gilt die Vorgabe der aktuellen Konfiguration</p> <table border="0" style="width: 100%;"> <tr> <td style="width: 30%;">ON</td> <td>Anfordern der Position unabhängig von der aktuellen Konfiguration</td> </tr> <tr> <td>OFF</td> <td>'Keine Positionsanforderung' unabhängig von der aktuellen Konfiguration</td> </tr> <tr> <td>USE_ACTUAL</td> <td>Es gilt die Vorgabe der aktuellen Konfiguration</td> </tr> </table>	ON	Anfordern der Position unabhängig von der aktuellen Konfiguration	OFF	'Keine Positionsanforderung' unabhängig von der aktuellen Konfiguration	USE_ACTUAL	Es gilt die Vorgabe der aktuellen Konfiguration
ON	Anfordern der Position unabhängig von der aktuellen Konfiguration						
OFF	'Keine Positionsanforderung' unabhängig von der aktuellen Konfiguration						
USE_ACTUAL	Es gilt die Vorgabe der aktuellen Konfiguration						

SetDefaultConfig =..	<p>Initialisieren der Decoder Arbeitsdaten bei Programmstart. Siehe Kanalbetriebsart- SUPPRESS_PROG_START_INIT Wird SetDefaultConfig nicht genutzt, so gilt die Vorgabe der aktuellen Konfiguration.</p> <p>ON Initialisieren der Decoder Arbeitsdaten OFF Initialisierung der Decoder Arbeitsdaten abgewählt USE_ACTUAL Es gilt die Vorgabe der aktuellen Konfiguration</p>
ReportSceneSample =..	<p>Freischalten der Schnittstelle zum Protokollieren der Szenedaten. Siehe Kanalbetriebsart- BEARB_MODE_SCENE Wird ReportSceneSample nicht genutzt, so gilt die Vorgabe der aktuellen Konfiguration.</p> <p>ON Szenedaten protokollieren OFF Szenedatenprotokoll abschalten USE_ACTUAL Es gilt die Vorgabe der aktuellen Konfiguration</p>
ReportRunTimeMeasure =..	<p>Freischalten der Schnittstelle zum Protokollieren von Zeitstempeln. Siehe Kanalbetriebsart- ONLINE_PROD_TIME Wird ReportRunTimeMeasure nicht genutzt, so gilt die Vorgabe der aktuellen Konfiguration.</p> <p>ON Zeitstempeldaten erzeugen OFF Protokollierung der Zeitstempeldaten abschalten USE_ACTUAL Es gilt die Vorgabe der aktuellen Konfiguration</p>
ReportAxesPositionSample =..	<p>Freischalten der Schnittstelle zum Protokollieren von Achspositionen. Siehe Kanalbetriebsart- ON_LINE Wird ReportAxesPositionSample nicht genutzt, so gilt die Vorgabe der aktuellen Konfiguration.</p> <p>ON Protokolldaten der Achspositionen erzeugen OFF Protokollierung der Achspositionen abschalten USE_ACTUAL Es gilt die Vorgabe der aktuellen Konfiguration</p>



Programmierbeispiel

Starten eines NC-Programms in einem anderen Kanal

Start des CNC Programms JM-1-ch2.nc in einem Kanal mit der log_id "1" ohne Positionsanforderung und ohne Initialisierung des Dekoders. Der Übergabeparameter @PL5 kann im auftragnehmenden Kanal ausgelesen werden, alle anderen @Plx nicht. Beim Leseversuch entsteht ein Fehler.

```
%ExampleMC_MovePath1 Master
N010 #MC_MovePath [CH=1, JobID = 5, FileName = "JM-1-ch2.nc", \
    InitializeOnActualPosition = OFF, SetDefaultConfig = OFF \
    @PL2 = 1000.5 @PL5 = 50]
N020 M30
```

In Programm JM-1-ch2.nc kann auf Parameter zugegriffen werden.

```
%ExampleMC_MovePath1 Slave „JM-1-ch2.nc“
N010 V.L.Parameter1 = @PL5 ;V.L.Parameter1 = 50
N020 X@PL2 ;Commanded X Position 1000.5
;...
```



Hinweis

Nach Einlesen des `#MC_MovePath` wird das Interpretieren des CNC Programms erst dann fortgesetzt, wenn der mit dem Befehl kommandierte Job erfolgreich in der Job-Warteschlange des auftragnehmenden Kanals abgelegt werden konnte.

3.3 Sperren konkurrierender Auftraggeber

Mit dem Ausführen des #LOCK Befehls erhält ein Auftraggeber exklusiven Zugriff. Der sperrende Zugriff für alle andere Auftraggeber (der gleichen Jobmanager-Gruppe) Zugriff wird mit #UNLOCK aufgehoben.

Es ist nicht zulässig innerhalb eines gelockten Blocks weitere #LOCK einzusetzen bzw. außerhalb des Lock-Blocks weitere #UNLOCK Befehle zu verwenden.

Ist bei Hauptprogrammende (M30) noch ein Lock-Block aktiv, so wird dieser implizit beendet und eine Warnung ausgegeben.

Syntax:

#LOCK	Exklusiven Zugriff aktivieren
#UNLOCK	Exklusiven Zugriff aufheben



Hinweis

Der Lock-Block wirkt nur auf Jobmanager-spezifische Kommandos. Alle anderen CNC Kommandos werden unabhängig davon weiterhin ausgeführt.

Eigenschaften des #LOCK-Befehls:

Jeder neu eintreffende #LOCK erhält steuerungsintern einen initialen Zeitstempel. Über diesen Zeitstempel erfolgt die Priorisierung mehrerer (evtl. schon wartender) #LOCK Anforderungen. Der Kanal mit dem ältesten #LOCK Zeitstempel erhält den Zugriff und die Sperreigenschaft. Steht nach einem #UNLOCK ein weiterer #LOCK an, greift erneut die Zeitstempelpriorisierung und aktiviert den nächsten ältesten #LOCK.

Bei Zeitgleichheit wird der Kanal bevorzugt, der die kleinste Konfigurationsnummer „j“ im Parameter *master[j].log_id* (P-STUP-00206 [▶ 44]).

3.4 Warten auf Zustand von beauftragtem Job

Mit dem #WAIT-Befehl synchronisiert sich ein auftraggebendes Programm mit einem zuvor von ihm kommandierten (beauftragten) Job. Das auftraggebende Programm wird bis zum Eintritt des erwarteten Zustands angehalten. Werden mehrere alternative Zustände angegeben, so reicht das Erfüllen eines der angefragten Zustände aus, das Programm weiterlaufen zu lassen. Über die Funktion MCV_WAIT_STATUS kann der letzte gültige Rückgabewert von #WAIT MC_Status ausgelesen und für die weitere Programmausführung genutzt werden.

Verfügbar ab V3.1.3110

Syntax NC-Befehl:

```
#WAIT MC_Status [ JobID=.. [MC_NEW] [MC_BUSY] [MC_ACTIVE] [MC_DONE]
                  [MC_ABORTED] [MC_ERROR] ]
```

JobID=..	Nutzerspezifische JobID/ Auftrags-Identifikationsnummer. Jede Nummer muss eindeutig innerhalb des auftraggebenden Programms sein.
MC_NEW	Neuer Auftrag erhalten, welcher im beauftragten Kanal noch nicht weiterverarbeitet wurde.
MC_BUSY	Neuer Auftrag wartet auf die Ausführung im beauftragten Kanal.
MC_ACTIVE	Der Auftrag ist aktiv im beauftragten Kanal, wurde jedoch noch nicht beendet.
MC_DONE	Der Auftrag wurde im beauftragten Kanal erfolgreich beendet.
MC_ABORTED	Der Auftrag wurde im beauftragten Kanal vor der Vollendung abgebrochen. Ein Abbruch kann durch einen Reset-Befehl z.B. #MC_GroupResetForced erfolgen.
MC_ERROR	Ein Fehler trat während der Bearbeitung des Auftrags auf.

Eigenschaften:

- Job-IDs werden mit dem Aufruf von #WAIT MC_Status in einer Historie gespeichert, um auch nach Beendigung/Abbruch des Jobs bei einer „späteren“ Abfrage einen korrekten Status zurückliefern zu können. Ist die in JobID angegebene Job-ID unbekannt, da nie kommandiert oder nicht mehr in der Historie, so wird ein Fehler ausgegeben.
- Die Job-ID Historie wird mit dem Beenden des Programms gelöscht. Es kann daher nicht mehr auf Jobs vorhergegangener (Auftraggeber-) Jobs synchronisiert werden. Es gibt auch keine Möglichkeit, sich mit Jobs eines anderen Auftraggebers zu synchronisieren.



Hinweis

Wird bei #WAIT MC_Status kein erwarteter Status definiert, so gilt ein implizites MC_DONE und MC_ABORTED.



Programmierbeispiel

Warten auf Jobzustand

Das Programm startet mit Job 633 ein Programm „SlaveFile“.nc im logischen Kanal 3, wartet dann auf die Erfüllung des Jobs mit den alternativen Zuständen MC_DONE, MC_ABORTED oder MC_ERROR. Wird keiner der Zustände erreicht, bleibt das Programm stehen.

```
%Example MC_Wait
N010 #MC_MovePath SYN[ CH=3 JobID=633 FileName="SlaveFile.nc"]
N020 #WAIT MC_Status [JobID=633 MC_DONE MC_ABORTED MC_ERROR]
N100 M30
```

3.5 Abfrage des zuletzt gültigen Status in #WAIT MC_Status

Nach einem oder mehreren vorausgegangen #WAIT MC_Status [▶ 24] Befehlen kann über die Funktion MCV_WAIT_STATUS der letzte gültige Rückgabewert von #WAIT MC_Status ausgelesen und für die weitere Programmausführung genutzt werden.

Als Rückgabewerte gelten die in den Konstanten hinterlegten Zahlenwerte.

Im CNC Code sind die folgenden Konstanten für den Status verfügbar:

- MC_NEW
- MC_BUSY
- MC_ACTIVE
- MC_DONE
- MC_ERROR
- MC_ABORTED

Diese dienen zum Vergleich einer gespeicherten Variablen nach einem #MCV_WAIT_STATUS.



Versionshinweis

Verfügbar ab V3.1.3110

Syntax:

#MCV_WAIT_STATUS [JobID=..]

JobID=.. Nutzerspezifische JobID/ Auftrags-Identifikationsnummer. Die Nummer ist eindeutig innerhalb des auftraggebenden Programms.



Hinweis

Der Aufruf der Funktion setzt für die in <JobID> hinterlegte Nummer einen oder mehrere #WAIT MC_Status Befehle mit der gleichen JobID=.. voraus.

Trifft dies nicht zu, so wird ein Fehler ausgegeben.



Programmierbeispiel

Warten auf Signal eines Auftrags

Das aufgeführte Programm startet mit Job 633 ein Programm „SlaveFile“.nc im logischen Kanal 3, wartet dann auf die Erfüllung des Jobs mit den alternativen Zuständen MC_DONE, MC_ABORTED oder MC_ERROR.

Der mit dem Weiterschalten des #WAIT MC_Status gültige Zustand kann ausgewertet werden.

```
%Example MC_Wait
N010 #MC_MovePath SYN[ CH=3 JobID=633 FileName="SlaveFile.nc"]
```

```

N010 #WAIT MC_Status [JobID=633 MC_DONE MC_ABORTED MC_ERROR]
N020 V.P.McStatus = MCV_WAIT_STATUS [633]
N030 $IF V.P.McStatus != MC_DONE
N040 #ERROR [ID455 MID0 RC2 PV1=V.P.McStatus \
          PV2=MC_DONE PM1=3 PM2=633]
N050 $ENDIF
N100 M30
    
```

3.6 Kanalreset in einem beauftragten Kanal

Der Befehl `MCV_GrpResetForced` kommandiert einen Kanalreset in einem anderen Kanal.



Versionshinweis

Verfügbar ab V3.1.3110

Syntax:

#MCV_GrpResetForced [SYN] [CH=.. JobID=..]

- SYN** Synchrone Befehlsausführung des ISO Programms. Vor der Ausführung des Befehls erfolgt eine implizite Kanalsynchronisation (implizites `#FLUSH WAIT`).
Ohne SYN wird der Befehl ohne Kanalsynchronisation ausgeführt.
- CH=..** Logische Kanalnummer P-STUP-00208 [▶ 45] des Kanals, in dem der Auftrag ausgeführt werden soll.
- JobID=..** Nutzerspezifische Auftrags Identifikationsnummer (JobID). Jede Nummer muss eindeutig innerhalb des im Masterkanal laufenden auftraggebenden Programms sein.
Diese JobID wird z.B. im `#WAIT MC_Status [▶ 24]` Befehl für die Auftragsidentifizierung verwendet.



Programmierbeispiel

Durchführen eines Resets in einem anderen Kanal

Das aufgeführte Programm setzt den Kanal 3 zurück.

```

%ExampleMCV_GrpResetForced
N010 N030 #MCV_GrpResetForced SYN [ CH=3 ID=634]
N020 M30
    
```

3.7 V.G.Variablen

In Verbindung mit dem Jobmanager wird die Statusvariable **V.G.IP_NR** belegt. Sie ist ausschließlich lesbar. Der Wert der Variablen ist die logische Nummer des auftraggebenden Kanals.

In einem NC-Programm eines auftraggebenden Kanals gibt `V.G.IP_NR` die eigene logische Kanalnummer zurück. Diese ist in der Konfiguration des Jobmanagers in P-STUP-00206 [▶ 44] festgelegt.

Wird die Variable in einem auftragnehmenden Kanal eingesetzt, so liefert sie, wenn der Job von einem auftraggebenden Job aus gestartet wurde, die logische Kanalnummer des auftraggebenden Kanals.

Wurde das Programm direkt im auftragnehmenden Kanal gestartet, so wird der Wert von P-STUP-00208 [▶ 45] aus der Konfiguration zurückgegeben.

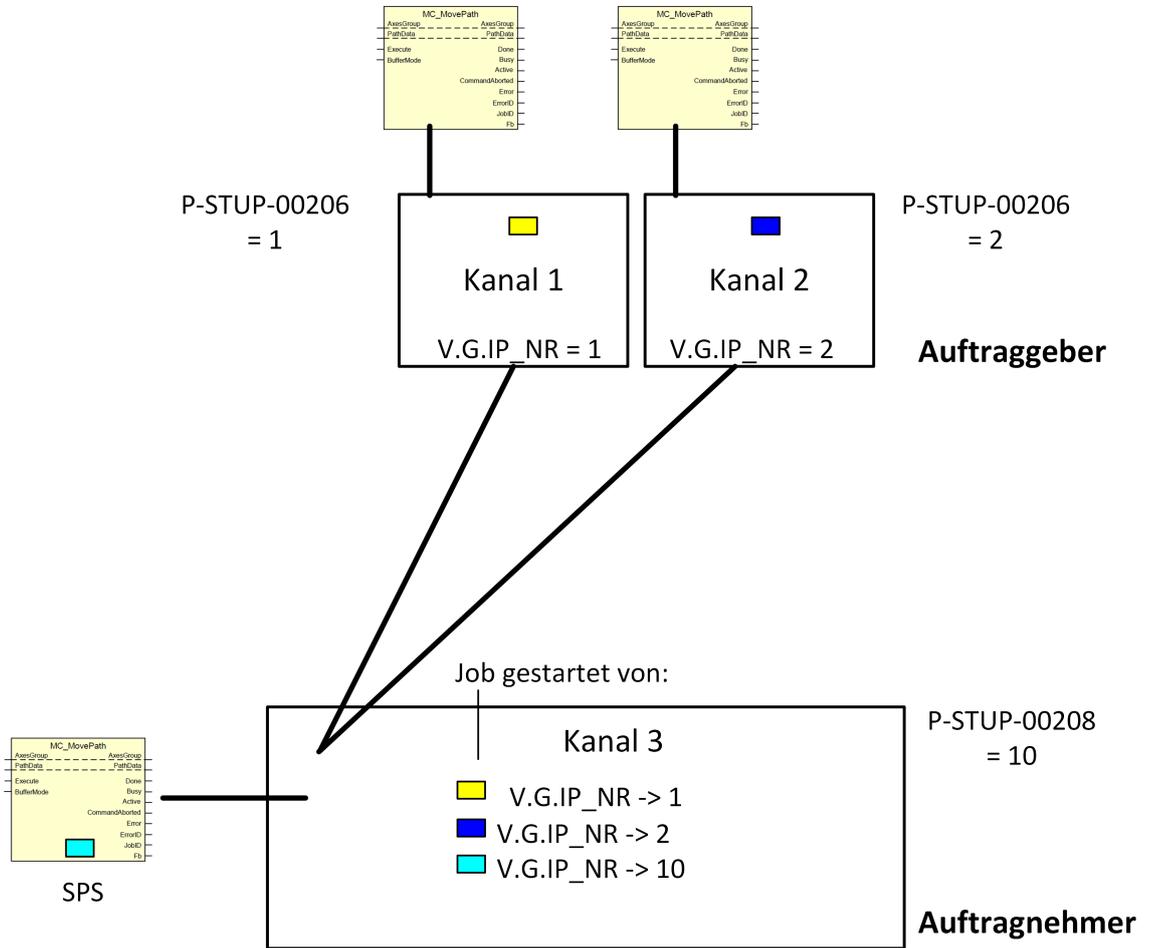


Abb. 6: Werte der V.G.Variable V.G.IP_NR im Kontext von auftraggebendem und auftragnehmenden Kanal

4 CNC-Objekte

Die nachfolgend beschriebenen CNC Objekte sind über die Task COM erreichbar.

Sie bilden die Schnittstellen von SPS Bausteinen nach. Die Tabellen beschreiben die jeweiligen Initialwerte, die bei Bedarf überschrieben werden können und die jeweils zugehörigen Eingangs-Pins der SPS Bausteine.

4.1 MC_MovePath + MCV_GrpPathPrepare + MCV_GrpPathMode

Objekt	Initialwert	Entsprechendes Element bzw. Konstante in PLCopen
mc_plco_move_path_file_name_w	""	MCV_GrpPathPrepare >FileName
mc_plco_move_path_fb_id_w	HMI	-
mc_plco_move_path_jstack_count_w	0	2, wenn (0) und (1) beschrieben sind 1, wenn nur (0) gültige Daten enthält 0, wenn keine JobID Daten gültig
mc_plco_move_path_jstack_id_0_w	0	MCV_GrpPathPrepare >JobID.JobID (0)
mc_plco_move_path_jstack_index_0_w	0	MCV_GrpPathPrepare
mc_plco_move_path_jstack_id_1_w	0	MCV_GrpPathPrepare >JobID.JobID (0)
mc_plco_move_path_jstack_index_1_w	0	MCV_GrpPathPrepare >JobID.IfclD.i_index [Source-ID (1)]
mc_plco_move_path_parameter_w	""	MCV_GrpPathPrepare >PathParameter
mc_plco_move_path_coord_sys_w	0	-
mc_plco_move_path_buffer_mode_w	0	-
mc_plco_move_path_trans_mode_w	0	-
mc_plco_move_path_init_on_act_pos_w	USE_ACTUAL	MCV_GrpPathMode >InitializeOnActualPosition
mc_plco_move_path_set_def_config_w	USE_ACTUAL	MCV_GrpPathMode >SetDefaultConfig
mc_plco_move_path_rep_scene_samp_w	USE_ACTUAL	MCV_GrpPathMode >ReportSceneSample
mc_plco_move_path_rep_ax_pos_samp_w	USE_ACTUAL	MCV_GrpPathMode >ReportAxesPositionSample

Das Schreiben von mc_plco_move_path_file_name_w löst den Befehl aus, die anderen Objekte können zuvor in beliebiger Reihenfolge geschrieben werden.

4.2 MCV_GrpResetForce

Objekt	Initialwert	Entsprechendes Element bzw. Konstante in PLCopen
mc_plco_reset_w	""	1
mc_plco_reset_fb_id_w	HMI	-
mc_plco_reset_jstack_count_w	0	2, wenn (0) und (1) beschrieben sind 1, wenn nur (0) gültige Daten enthält 0, wenn keine JobID Daten gültig
mc_plco_reset_jstack_id_0_w	0	JobID.JobID (0)
mc_plco_reset_jstack_index_0_w	0	JobID.IfclD.i_index [Source-ID (0)]
mc_plco_reset_jstack_id_1_w	0	JobID.JobID (1)
mc_plco_reset_jstack_index_1_w	0	JobID.IfclD.i_index [Source-ID (1)]

Das Schreiben von mc_plco_reset_w löst den Befehl aus, die anderen Objekte können zuvor in beliebiger Reihenfolge geschrieben werden.

4.3 MCV_GrpReadJobAck

Objekt	Entsprechendes Element bzw. Konstante in PLCopen
mc_plco_resp_job_state_r	0: keine Daten gelesen 1: Daten gelesen und gültig
mc_plco_resp_fb_id_r	Aktueller Wert des HMI Interface
mc_plco_resp_error_id_r	MCV_GrpReadJobAck>ErrorID
mc_plco_resp_jstack_count_r	2, wenn (0) und (1) beschrieben sind 1, wenn nur (0) beschrieben ist 0, wenn kein Parameter gültig ist
mc_plco_resp_jstack_id_0_r	MCV_GrpReadJobAck>JobID.JobID (0)
mc_plco_resp_jstack_index_0_r	MCV_GrpReadJobAck>JobID.IfclD [Source-ID (0)]
mc_plco_resp_jstack_id_1_r	MCV_GrpReadJobAck>JobID.JobID (1)
mc_plco_resp_jstack_index_1_r	MCV_GrpReadJobAck>JobID.IfclD [Source-ID (1)]

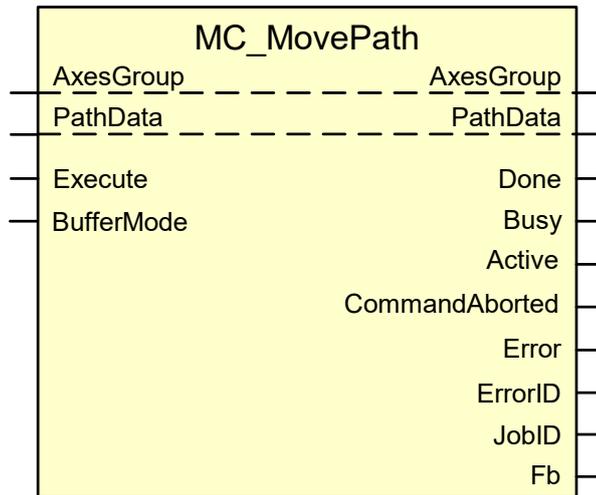
Das Lesen von mc_plco_resp_job_state_r löst den Lesevorgang aus, danach haben die anderen Objekte jeweils ihren Wert.

5 PLCopen-Funktionsbausteine

5.1 MC_MovePath

Die Implementierung dieses Funktionsbausteins ermöglicht die Ausführung eines NC-Programmes. Der Name der Datei, die das NC-Programm enthält, wird in der Datenstruktur MC_PATH_DATA_REF angegeben.

Blockdiagramm



Parameter des FB

VAR_IN_OUT		
Variablenname	Datentyp	Beschreibung
AxesGroup	AXES_GROUP_REF	Achsgruppenreferenz
PathData	MC_PATH_DATA_REF	Referenz auf die Bahnbeschreibung. In dieser Implementierung ist damit ein NC-Programm gemeint.
VAR_INPUT		
Variablenname	Datentyp	Beschreibung
Execute	BOOL	Startet die Beauftragung mit steigender Flanke.
BufferMode	MC_BUFFER_MODE	Der Eingang legt fest, wann die Aktivierung eines Auftrags stattfindet, wenn bei der Beauftragung des FB bereits andere Aufträge aktiv sind oder auf die Ausführung warten. Außerdem wird damit die Bahnerzeugung am Übergang zwischen 2 Aufträgen festgelegt. Folgende Werte sind möglich: mcAborting = 0 mcBuffered = 1 mcBlendingPrev = 3
VAR_OUTPUT		
Variablenname	Datentyp	Beschreibung
Done	BOOL	Die Ausführung des NC-Programmes ist abgeschlossen.
Busy	BOOL	TRUE zeigt an, dass der FB mit einem Auftrag beschäftigt ist.
Active	BOOL	Die Beauftragung des FB ist noch nicht abgeschlossen.
CommandAborted	BOOL	Die Ausführung des NC-Programmes wurde unterbrochen.
Error	BOOL	TRUE zeigt an, dass ein Fehler aufgetreten ist.
ErrorID	WORD	Fehlerkennung
JobID	UDINT	Ordnungsnummer des letzten über den FB abgesetzten Auftrages.
Fb	MCV_FB_REF	Optional für die Nutzung in Applikationen mit Auftragsverwaltung (MCV_GrpReadJobAck) [▶ 37] Dieser Ausgang ist ab der Version CNC-Version V3.1.3103.1 verfügbar.

5.2 MCV_GrpPathMode

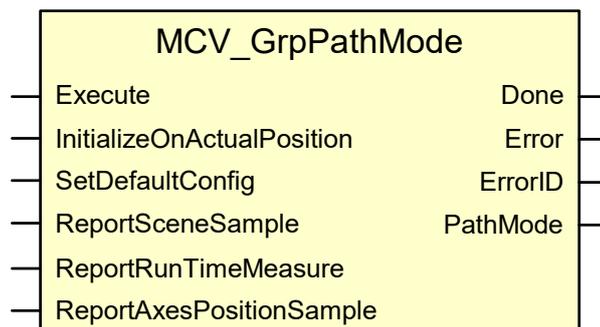


Versionshinweis

Dieser Baustein ist verfügbar ab der CNC-Version V3.1.3103.1

Der Baustein dient dazu, die Optionen bei der Ausführung eines zu startenden Auftrags in die Datenstruktur MCV_PATH_MODE des Ausgangs "PathMode" einzutragen. Der Ausgang wird mit dem "PathMode"-Eingang einer Instanz vom Typ MCV_GrpPathPrepare [► 34] verbunden.

Blockdiagramm



Parameter des FB

VAR_INPUT		
Variablenname	Datentyp	Beschreibung
Execute	BOOL	Bei steigender Flanke am Eingang werden die Werte der Eingangspins, die sich auf die Startoptionen beziehen, in eine Datenstruktur vom Typ PATH_MODE eingetragen und am Ausgang PathMode ausgegeben.
InitializeOnActualPosition	MCV_START_MODE	
SetDefaultConfig	MCV_START_MODE	
ReportSceneSample	MCV_START_MODE	
ReportRunTimeMeasure	MCV_START_MODE	
ReportAxesPositionSample	MCV_START_MODE	
VAR_OUTPUT		
Variablenname	Datentyp	Beschreibung
Done	BOOL	TRUE zeigt an, dass die Werte der Startoptionen am Ausgang PathMode in einer Struktur vom Typ MCV_PATH_MODE verfügbar sind.
Error	BOOL	TRUE zeigt an, dass ein Fehler aufgetreten ist.
ErrorID	WORD	Fehlerkennung
PathMode	MCV_PATH_MODE	Parameter für Programmmodus eines Auftrags

5.3 MCV_GrpPathPrepare



Versionshinweis

Dieser Baustein ist verfügbar ab der CNC-Version V3.1.3103.1

Der Funktionsblock wird eingesetzt, um Auftragsdaten, Programmparameter, und Programmmodi in eine Datenstruktur vom Typ MC_PATH_DATA_REF zu übertragen. Diese Struktur wird am Ausgang PathData ausgegeben und ist an den entsprechenden Eingang einer MC_MovePath [► 30] Instanz zu übergeben.

Die Eingänge des Bausteins können vom Anwender selbst beschrieben werden. Alternativ kann der Baustein an die Ausgänge einer MCV_GrpGetJobRequest Instanz angebunden werden um z.B. von einer HMI oder einem anderen NC-Kanal erzeugte Aufträge weiterreichen zu können.



Hinweis

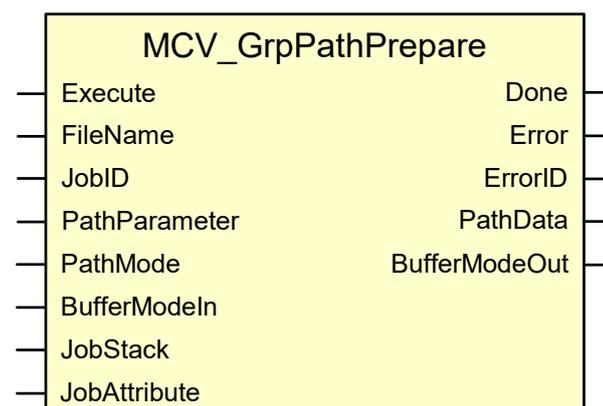
Die Struktur MC_PATH_DATA_REF darf nur von Instanzen des Typs MCV_GrpPathPrepare beschrieben werden. Dies gewährleistet, dass zukünftige Änderungen der Bibliothek keine Auswirkungen auf vorhandene SPS Programme haben.



Hinweis

Zulässig bleibt weiterhin das direkte Eintragen eines Namens bzw. Pfads für ein NC-Programm auf MC_PATH_DATA_REF. Damit können bestehende SPS- Applikationen ohne Auftragsverwaltung oder Parametrierung unverändert verwendet werden. Der Einsatz von MCV_GrpPathPrepare ist dann nicht zulässig.

Blockdiagramm



Parameter des FB

VAR_INPUT		
Variablenname	Datentyp	Beschreibung
Execute	BOOL	Bei steigender Flanke am Eingang wird der Baustein ausgeführt.
FileName	STRING(MCV_PROG_NAME_STRLEN)	Programmname eines Auftrags.
JobID	MCV_GRP_JOB_ID	Jobinformation eines Auftrags
PathParameter	MCV_PATH_PARAM	<p>Programmparameter eines Auftrags.</p> <p>Im aufgerufenen NC-Programm (FileName) kann auf die Parameter zugegriffen werden. PL[0] entspricht der Variablen @PL1, PL[19] der Variablen @PL20. Ungenutzte Parameter sind mit "0" vorinitialisiert.</p>
PathMode	MCV_PATH_MODE	<p>Parameter für Programmmodus eines Auftrags. Verfügbar sind</p> <ul style="list-style-type: none"> - InitializeOnActualPosition, - SetDefaultConfig, - ReportSceneSample, - ReportRunTimeMeasure, - ReportAxesPositionSample mit <p>0 = HLI_MC_START_MODE_NOT_USED 1 = HLI_MC_START_MODE_OFF 2 = HLI_MC_START_MODE_ON 3 = HLI_MC_START_MODE_USE_ACTUAL</p>
BufferModeIn	MC_BUFFERMODE	<p>BufferMode eines Auftrags.</p> <p>Folgende Werte sind (analog zur Schnittstelle BufferMode im Baustein MC_MovePath [► 30]) möglich:</p> <p>mcAborting= 0 mcBuffered= 1 mcBlendingPrev= 3</p>
JobStack	HLI_MC_JOB_STACK	Information zur Quelle des Auftrags. Nur benötigt, wenn der Auftrag von einer Instanz des MCV_GrpGetJobRequest eingelesen wurde. Ansonsten sollte der Eingang unbelegt bleiben.
JobAttribute	UDINT	Identifikationsnummer, die vom Anwender frei vergeben werden kann.
VAR_OUTPUT		
Variablenname	Datentyp	Beschreibung
Done	BOOL	TRUE zeigt an, dass die Datenstruktur MC_PATH_DATA_REF korrekt ausgefüllt wurde und gemeinsam mit dem Wert des Ausgangs "BufferModeOut" an eine Instanz des MC_MovePath [► 30] übergeben werden kann.
Error	BOOL	TRUE zeigt an, dass ein Fehler aufgetreten ist.
ErrorID	WORD	Fehlerkennung

PathData	MC_PATH_DATA_REF	Struktur zur Verwendung an dem Eingang "PathData" einer Instanz des MC_MovePath [► 30].
BufferModeOut	MC_BUFFERMODE	Der BufferMode zur Verwendung an dem Eingang "BufferMode" einer Instanz des MC_MovePath.

5.4 MCV_GrpReadJobAck



Versionshinweis

Dieser Baustein ist verfügbar ab der CNC-Version V3.1.3103.1

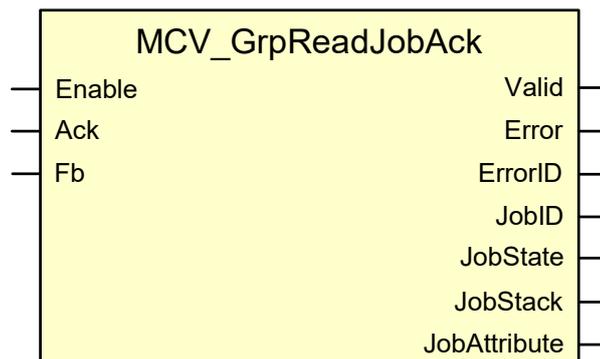
Der Funktionsblock wird eingesetzt, um die Beendigung bzw. den Abbruch von Aufträgen auswerten zu können, die über einen MC_MovePath [▶ 30] kommandiert wurden. Die Instanz des MC_MovePath [▶ 30], deren Auftragsrückmeldungen ausgewertet werden, identifiziert sich über den Eingang "Fb".



Hinweis

Wenn der Baustein angelegt und der Eingang "Enable" auf TRUE gesetzt ist, muss der Baustein zyklisch aufgerufen werden. Ansonsten werden Auftragsrückmeldungen zu beliebigen – auch anderen PLCopen Part4 Bausteinen – blockiert!

Blockdiagramm



Parameter des FB

VAR_INPUT		
Variablenname	Datentyp	Beschreibung
Enable	BOOL	Solange "Enable" TRUE ist werden Rückmeldungen von Aufträgen gelesen. Es werden nur Rückmeldungen angezeigt, die sich auf Aufträge beziehen, die von der am Eingang "Fb" angemeldeten Instanz eines MC_MovePath [▶ 30] abgesetzt wurden. Es werden nur dann neue Rückmeldungen angezeigt, wenn der Eingang "Ack" FALSE ist.
Ack	BOOL	Der Eingang legt fest, dass die an den Ausgängen des Bausteins angezeigten Informationen gelesen wurden.
Fb	MCV_FB_REF	Verweis auf eine Instanz eines MC_MovePath [▶ 30] Bausteins mit dem Ausgang des gleichen Typs.
VAR_OUTPUT		
Variablenname	Datentyp	Beschreibung
Valid	BOOL	TRUE zeigt an, dass neue Daten zur Abholung bereit liegen.
Error	BOOL	TRUE zeigt an, dass ein Fehler aufgetreten ist.
ErrorID	WORD	Fehlerkennung
JobID	MCV_GRP_JOB_INFO	Beschreibung des Jobs und des Status der Beauftragung.
JobState	MCV_JOB_STATE	Jobinformation eines Auftrags
JobStack	HLI_MC_JOB_STACK	Information zur Quelle des Auftrags. Nur benötigt, wenn der Auftrag zu einer Instanz des MCV_GrpSetJobResponse weitergereicht werden soll.
JobAttribute	UDINT	Identifikationsnummer, die vom Anwender vorgegeben wurde und bei der Ausgabe des Zustands eines Jobs mit ausgegeben wird.

5.5 MCV_GrpReadJobStatusList

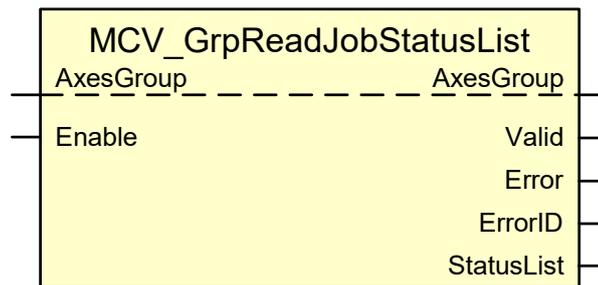


Versionshinweis

Dieser Baustein ist verfügbar ab der CNC-Version V3.1.3103.1

Der Funktionsblock wird eingesetzt, um die Stati aller aktuell für eine Achsgruppe kommandierten Aufträge anzuzeigen.

Blockdiagramm



Parameter des FB

VAR_IN_OUT		
Variablenname	Datentyp	Beschreibung
AxesGroup	AXES_GROUP_REF	Achsgruppenreferenz
VAR_INPUT		
Variablenname	Datentyp	Beschreibung
Enable	BOOL	Ist der Eingang TRUE so werden die Ausgänge des Bausteins aktualisiert.
VAR_OUTPUT		
Variablenname	Datentyp	Beschreibung
Valid	BOOL	TRUE zeigt an, dass die angeforderte Aktualisierung erfolgte.
Error	BOOL	TRUE zeigt an, dass ein Fehler aufgetreten ist.
ErrorID	WORD	Fehlerkennung
StatusList	MCV_GRP_JOB_DISPLAY	Struktur, die die Stati aller aktuellen MC_MovePath [▶ 30]-Aufträge in einem Kanal beschreibt .

5.6 MCV_GrpResetForced



Versionshinweis

Dieser Baustein ist verfügbar ab der CNC-Version V3.1.3103.1

Der Funktionsbaustein führt zu einem gesteuerten Bewegungshalt. Er bricht jede laufende Beauftragung durch andere Bewegungs-FB ab.

Die Achsgruppe wechselt in den Zustand „GroupStopping“, bis die Geschwindigkeit 0 erreicht ist. Wenn der Ausgang „Done“ und „Execute“ auf FALSE gesetzt wird, wechselt Achsgruppenzustand nach „GroupStandby“.

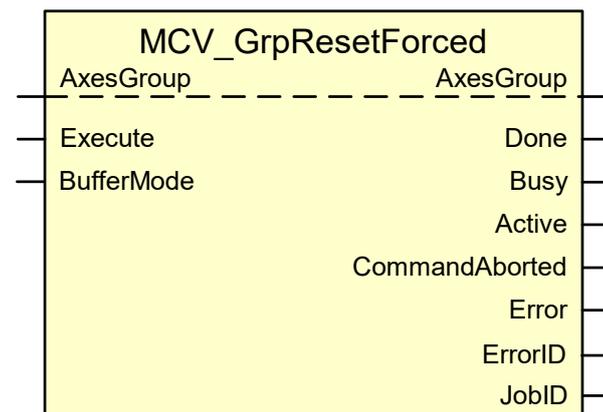
Der Funktionsbaustein ist nicht unterbrechbar.



Hinweis

Zusätzlich zum Bewegungshalt, der identisch zum MC_GrpStop [► 42] implementiert ist, versetzt dieser FB die Achsgruppe in den Grundzustand.

Blockdiagramm



Parameter des FB

VAR_IN_OUT		
Variablenname	Datentyp	Beschreibung
AxesGroup	AXES_GROUP_REF	Achsgruppenreferenz
VAR_INPUT		
Variablenname	Datentyp	Beschreibung
Execute	BOOL	Startet die Beauftragung mit steigender Flanke.
BufferMode	MC_BUFFER_MODE	Der Eingang legt fest, wann die Aktivierung eines Auftrags stattfindet, wenn bei der Beauftragung des FB bereits andere Aufträge aktiv sind oder auf die Ausführung warten. Folgende Modi werden unterstützt: mcAborting = 0 mcBuffered = 1
VAR_OUTPUT		
Variablenname	Datentyp	Beschreibung
Done	BOOL	TRUE zeigt an, dass die Geschwindigkeit 0 erreicht wurde. Die Achsen stehen still.
Busy	BOOL	TRUE zeigt an, dass der FB mit einem Auftrag beschäftigt ist.
Active	BOOL	Die Beauftragung ist noch nicht abgeschlossen.
CommandAborted	BOOL	Der Auftrag zum Anhalten wurde durch einen anderen Auftrag abgebrochen.
Error	BOOL	TRUE zeigt an, dass ein Fehler aufgetreten ist.
ErrorID	WORD	Fehlerkennung
JobID	UDINT	Ordnungsnummer des letzten über den FB abgesetzten Auftrages.

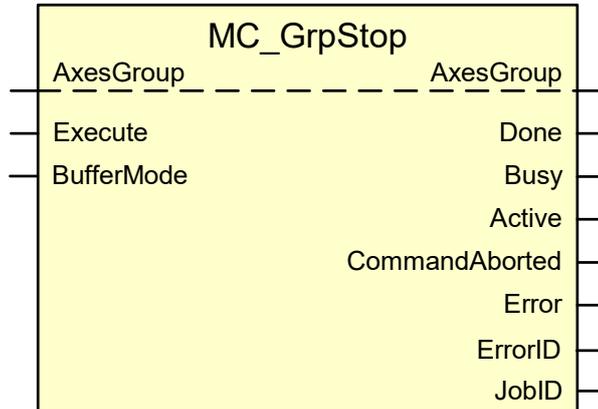
5.7

MC_GrpStop

Der Funktionsbaustein führt zu einem gesteuerten Bewegungshalt. Er bricht jede laufende Beauftragung durch andere Bewegungs-FB ab.

Die Achsgruppe wechselt in den Zustand „GroupStopping“, bis die Geschwindigkeit 0 erreicht ist. Wenn der Ausgang „Done“ und „Execute“ auf FALSE gesetzt wird, wechselt Achsgruppenzustand nach „GroupStandby“. Der Funktionsbaustein ist nicht unterbrechbar.

Blockdiagramm



Parameter des FB

VAR_IN_OUT		
Variablenname	Datentyp	Beschreibung
AxesGroup	AXES_GROUP_REF	Achsgruppenreferenz
VAR_INPUT		
Variablenname	Datentyp	Beschreibung
Execute	BOOL	Startet die Beauftragung mit steigender Flanke.
BufferMode	MC_BUFFER_MODE	Der Eingang legt fest, wann die Aktivierung eines Auftrags stattfindet, wenn bei der Beauftragung des FB bereits andere Aufträge aktiv sind oder auf die Ausführung warten. Folgende Modi werden unterstützt: mcAborting = 0 mcBuffered = 1
VAR_OUTPUT		
Variablenname	Datentyp	Beschreibung
Done	BOOL	TRUE zeigt an, dass die Geschwindigkeit 0 erreicht wurde. Die Achsen stehen still.
Busy	BOOL	TRUE zeigt an, dass der FB mit einem Auftrag beschäftigt ist.
Active	BOOL	Die Beauftragung ist noch nicht abgeschlossen.
CommandAborted	BOOL	Der Auftrag zum Anhalten wurde durch einen anderen Auftrag abgebrochen.
Error	BOOL	TRUE zeigt an, dass ein Fehler aufgetreten ist.
ErrorID	WORD	Fehlerkennung
JobID	UDINT	Ordnungsnummer des letzten über den FB abgesetzten Auftrages.

6 Parameter

P-STUP-00206	Logische ID eines auftraggebenden Kanals in einer Job Manager Gruppe
Beschreibung	Der Parameter bestimmt die logische ID eines auftraggebenden Kanals. Diese logische ID kann im auftragnehmenden Kanal durch V.G.IP_NR abgefragt werden. Wegen der benötigten Eindeutigkeit darf kein zweiter auftraggebender Kanal der gleichen Jobmanager-Gruppe die gleiche logische ID nutzen.
Parameter	jobmanager.group[i].master[j].log_id mit i = 0, 1 (Index der Jobmanager-Gruppe, max. 1) mit j = 0 ... n (Index eines fortlaufenden Listenelements. n: applikations-spezifisch)
Datentyp	UNS16
Datenbereich	1 ... 65536
Dimension	----
Standardwert	0
Anmerkungen	Verfügbar ab V3.1.3110 Wenn der Auftragnehmer nicht durch einen Auftraggeber beauftragt wurde, dann meldet der Aufruf von "V.G.IP_NR" die "log_id" des Auftragnehmers zurück!

P-STUP-00207	Kanal ID eines auftraggebenden Kanals in einer Jobmanager Gruppe
Beschreibung	Der durch den Parameter bezeichnete auftraggebende Kanal (Master) in einer Jobmanager-Gruppe entspricht einem vorhandenen CNC Kanal. Er kann keiner weiteren Gruppe zugeteilt werden, weder als "Auftragnehmer" (Slave) noch als "Auftraggeber" (Master). Die genutzte Nummer muss einer vorhandenen Kanalnummer entsprechen.
Parameter	jobmanager.group[i].master[j].channel_id (applikations-spezifisch) mit i = 0, 1 (Index der Jobmanager-Gruppe, max. 1) mit j = 0..n (Index eines fortlaufenden Listenelements. n: applikations-spezifisch)
Datentyp	UNS16
Datenbereich	1 ... 12 (applikationsspezifisch)
Dimension	----
Standardwert	0 *
Anmerkungen	Verfügbar ab V3.1.3110 * Der Wert entspricht der Aussage: Parameter wird nicht genutzt. Sind alle jobmanager.group[i].master[j].channel_id Parameter in einer Job Manager Gruppe "0", so ist der Job Manager für die Gruppe deaktiviert. Der auftraggebende Kanal entspricht einem "normalen" Kanal. Die Jobmanager Konfiguration erweitert den Befehlsvorrat um Jobmanager Befehle [► 19] (siehe [FCT-M10] [► 5]). Hierzu gehört auch, jeden in der gleichen Jobmanager Gruppe befindlichen Auftragnehmer (Slaves) mit Jobs beauftragen zu können

P-STUP-00208	Logische ID eines auftragnehmenden Kanals in einer Jobmanager Gruppe
Beschreibung	Der Parameter bestimmt die logische ID eines auftragnehmenden Kanals. Jeder Auftragnehmer (Slave) in einer Jobmanager Gruppe wird beim Beauftragen vom Auftraggeber mit seiner logischen ID "log_id" angesprochen. Wegen der benötigten Eindeutigkeit darf kein zweiter auftragnehmender Kanal der gleichen Jobmanager-Gruppe die gleiche logische ID nutzen.
Parameter	jobmanager.group[i].cnc_slave[j].log_id mit i = 0, 1 (Index der Jobmanager-Gruppe, max. 1) mit j = 0..n (Index eines fortlaufenden Listenelements. n: applikations-spezifisch)
Datentyp	UNS16
Datenbereich	1 ... 65536
Dimension	----
Standardwert	0
Anmerkungen	Verfügbar ab V3.1.3110 Es gibt zwei Typen von Auftragnehmern: CNC Kanäle und SPS Einheiten. Die logische ID bezieht sich immer auf den jeweiligen Typ

P-STUP-00209	Auftragnehmender Kanal in einer Job Manager Gruppe
Beschreibung	Der durch den Parameter bezeichnete auftragnehmende Kanal (Slave) in einer Jobmanager-Gruppe entspricht einem vorhandenen CNC Kanal. Er kann keiner weiteren Gruppe zugeteilt werden, weder als "Auftragnehmer" (Slave) noch als "Auftraggeber" (Master). Die genutzte Nummer muss einer vorhandenen Kanalnummer entsprechen.
Parameter	jobmanager.group[i].cnc_slave[j].channel_id (applikations-spezifisch) mit i = 0, 1 (Index der Jobmanager-Gruppe, max. 1) mit j = 0..n (Index eines fortlaufenden Listenelements. n: applikations-spezifisch)
Datentyp	UNS16
Datenbereich	1 ... (applikationsspezifisch)
Dimension	----
Standardwert	0
Anmerkungen	Verfügbar ab V3.1.3110 Ein auftragnehmender Kanal verhält sich wie ein "normaler" Kanal. Er hat die zusätzliche Eigenschaft, von einem beliebigen Master in der gleichen Job Manager Gruppe aufgefordert zu werden, einen Job auszuführen. Die Beendigung des Jobs wird steuerungsintern an den Auftraggeber zurückgemeldet.

P-STUP-00210	Logische ID einer auftragnehmenden SPS Einheit in einer Job Manager Gruppe
Beschreibung	Der Parameter bestimmt die logische ID einer auftragnehmenden SPS-Einheit. Jeder SPS Auftragnehmer (Slave) in einer Jobmanager Gruppe wird beim Beauftragen vom Auftraggeber mit seiner logischen ID "log_id" angesprochen. Wegen der benötigten Eindeutigkeit darf keine zweite auftragnehmender SPS-Einheit der gleichen Jobmanager-Gruppe die gleiche logische ID nutzen.
Parameter	jobmanager.group[i]. plc_slave[j].log_id mit i = 0, 1 (Index der Jobmanager-Gruppe, max. 1) mit j = 0...n (Index eines fortlaufenden HLI Listenelements. n: applikations-spezifisch)
Datentyp	UNS16
Datenbereich	1 ... 65536
Dimension	----
Standardwert	0
Anmerkungen	Verfügbar ab V3.1.3110

P-STUP-00211	Auftragnehmende SPS Einheit in einer Jobmanager-Gruppe
Beschreibung	Die durch den Parameter bezeichnete auftragnehmende SPS Einheit (Slave) in einer Jobmanager-Gruppe entspricht einer Schnittstelle auf dem HLI. Sie kann nach dieser Zuordnung keiner weiteren Gruppe zugeteilt werden.
Parameter	jobmanager.group[i].hli_index (applikations-spezifisch) mit i = 0, 1 (Index der Jobmanager-Gruppe, max. 1) mit j = 0..n (Index eines fortlaufenden HLI Listenelements n: applikationsspezifisch)
Datentyp	UNS08
Datenbereich	0 ... 31 (applikationsspezifisch)
Dimension	----
Standardwert	0
Anmerkungen	Verfügbar ab V3.1.3110 Eine auftragnehmende SPS Einheit hat die Eigenschaft, von einem beliebigen Master in der gleichen Jobmanager-Gruppe aufgefordert zu werden, einen Job auszuführen. Die Beendigung des Jobs wird steuerungsintern an den Auftraggeber zurückgemeldet.

P-STUP-00212	Parameterliste einer auftragnehmenden SPS Einheit in einer Jobmanager-Gruppe
Beschreibung	Die auftragnehmende SPS Einheit (Slave) in einer Jobmanager-Gruppe kann vom Auftraggeber beim Start Parameter erhalten. Die Datenstruktur beschreibt eine deklarierte Parameterliste der unter "jobmanager.coding[i].list" (P-STUP-00204). Der hier benötigte Parameter entspricht dem Index [i] aus "jobmanager.coding[i].list".
Parameter	jobmanager.group[i].plc_slave[j].coding mit i = 0, 1 (Index der Jobmanager-Gruppe, max. 1) mit j = 0..n (Index eines fortlaufenden HLI Listenelements n: applikations-spezifisch)
Datentyp	UNS08
Datenbereich	0 ... 2 (applikationsspezifisch)
Dimension	----
Standardwert	0
Anmerkungen	Verfügbar ab V3.1.3110

7 Anhang

Stichwortverzeichnis

D

D-MCP-P4- MCV_GrpReadJobAck	37
D-MCP-P4- MCV_GrpResetForced	40
D-MCP-P4-MC_GrpStop	42
D-MCP-P4-MC_MovePath	30
D-MCP-P4-MCV_GrpPathMode	32
D-MCP-P4-MCV_GrpPathPrepare	34
D-MCP-P4-MCV_GrpReadJobStatusList	39

M

MC_GrpStop	42
MC_MovePath	30
MCV_GrpPathMode	32
MCV_GrpPathPrepare	34
MCV_GrpReadJobAck	37
MCV_GrpReadJobStatusList	39
MCV_GrpResetForced	40

P

P-STUP-00206	44
P-STUP-00207	44
P-STUP-00208	45
P-STUP-00209	45
P-STUP-00210	46
P-STUP-00211	46
P-STUP-00212	47



© Copyright
ISG Industrielle Steuerungstechnik GmbH
STEP, Gropiusplatz 10
D-70563 Stuttgart
Alle Rechte vorbehalten
www.isg-stuttgart.de
support@isg-stuttgart.de

