

MCX – Motion-Control-Grundlagen

Application Note 054

608 860 65_00

Dieses Dokument wurde von der Jetter AG mit der gebotenen Sorgfalt und basierend auf dem ihr bekannten Stand der Technik erstellt. Änderungen und technische Weiterentwicklungen an unseren Produkten werden nicht automatisch in einem überarbeiteten Dokument zur Verfügung gestellt. Die Jetter AG übernimmt keine Haftung und Verantwortung für inhaltliche oder formale Fehler, fehlende Aktualisierungen sowie daraus eventuell entstehende Schäden oder Nachteile.

Jetter AG
Gräterstraße 2
71642 Ludwigsburg
Deutschland

www.jetter.de

Telefon:

Zentrale	+49 7141 2550-0
Vertrieb	+49 7141 2550-531
Technische Hotline	+49 7141 2550-444

E-Mail:

Technische Hotline	info@jetter.de
Vertrieb	hotline@jetter.de
	vertrieb@jetter.de

Produktname	MCX – Motion-Control-Grundlagen
Dokumentart	Application Note 054
Originaldokument	
Dokumentversion	1.00
Artikelnummer	608 860 65_00
Ausgabedatum	26.05.2021

Inhaltsverzeichnis

1	MCX vs. MotionControl	1
1.1	Objektmodell.....	1
1.2	Zustandsverwaltung.....	3
2	Konfiguration der MCX-Steuerung	6
2.1	Anlegen eines MCX-Projekts	6
2.1.1	Antriebsverstärker hinzufügen	7
2.1.2	Weitere MC-Objekte anlegen.....	9
2.1.3	Inkorrekte Hardwarekonfiguration	10
2.1.4	Konfiguration Achsobjekte	10
2.1.5	Konfiguration Achsverbunde.....	12
2.1.6	Konfiguration übertragen	13
3	Programmierung der MCX	15
3.1	Beispielprogramm.....	16
3.2	MotionAPI-Version wählen.....	16
3.2.1	MotionAPI hinzufügen.....	19

1 MCX vs. MotionControl

MotionControl ist eine Software, die die Bewegung einer Achse berechnet.

Meist läuft die MC-Software auf dem Servoverstärker. Im Einzelachsbetrieb werden damit die Positionen berechnet.

Die MCX ist dagegen eine zentrale Bewegungssteuerung. Die einzelnen Achsen werden über einen Feldbus angebunden und miteinander synchronisiert. Somit wird sichergestellt, dass alle Achsen zum gleichen Zeitpunkt einen neuen Sollwert erhalten.

Des Weiteren lassen sich hierdurch auch koordinierte Bewegungen der einzelnen Achsen erzeugen, wie z.B. elektrische Getriebe, Kurvenscheiben, Bahnverbunde.

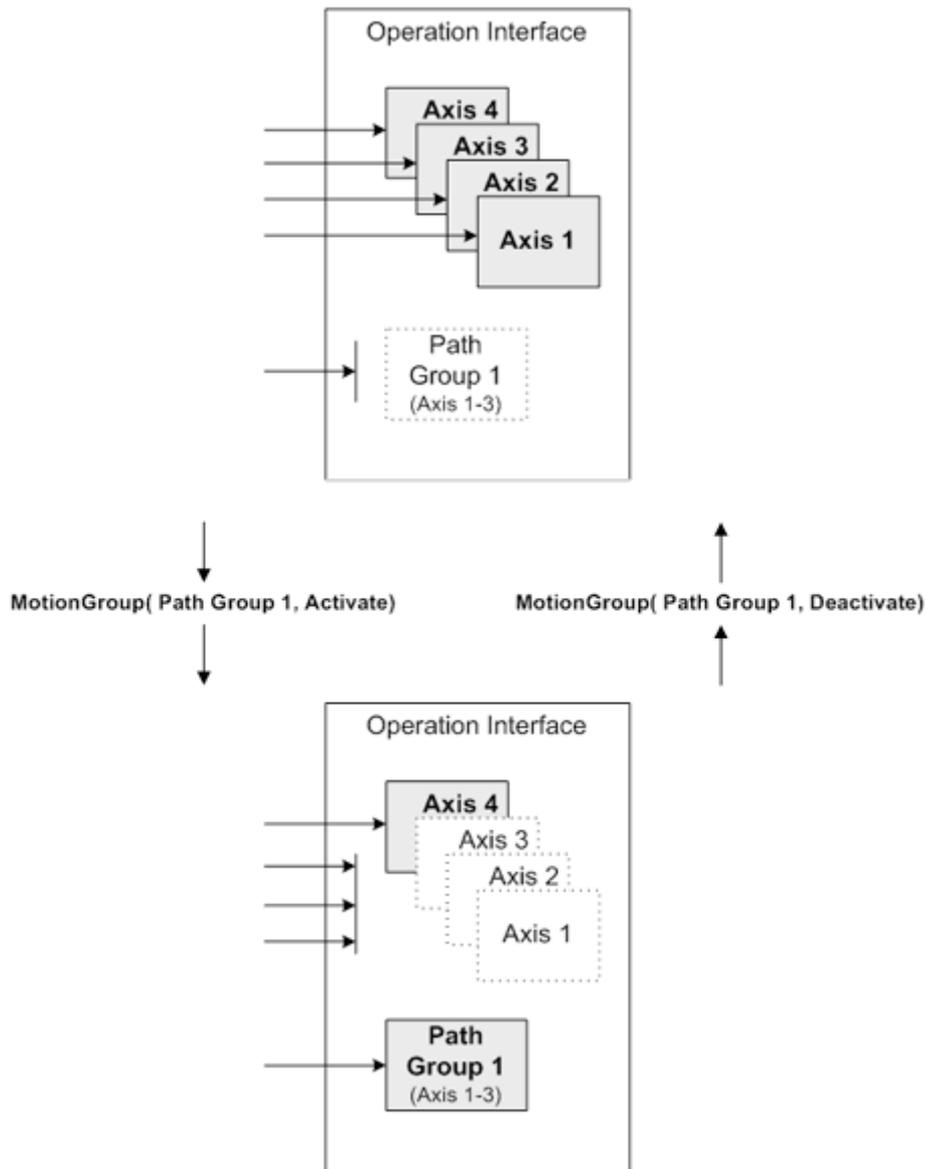
1.1 Objektmodell

In der MotionControl-Software werden einzelne Achsen oder Achsverbunde als Objekte gesehen. Diese Objekte werden zunächst im Hardware Manager angelegt und auch die Hardwarekonfiguration wird dort eingestellt. Zu den möglichen Objekten zählen:

- **Achsen** - sind in der Regel mit JetMove angetriebene, lineare oder rotatorische Einzelachsen, die als Normal- oder Modulo-Achsen betrieben werden können. Sie können auch als Simulationsachsen ausgelegt sein, falls noch keine Hardware angeschlossen ist. Es können aber auch rein virtuelle Achsen sein.
- **Bahnverbunde** - sind eine Zusammenfassung mehrerer zuvor konfigurierter Einzelachsen, die gemeinsam eine Bahn beschreiben. Meist bilden die beteiligten Achsen eine kinematische Kette, an deren Ende ein Werkzeug angebracht ist, über das die Achsen mechanisch gekoppelt sind. Beispiele hierfür sind 3D-kartesische Verbunde oder SCARA-Roboter etc.
- **Technologieverbunde** - sind eine Zusammenfassung mehrerer zuvor konfigurierter Einzelachsen, die gemeinsam einen Prozessablauf beschreiben. Meist besteht zwischen den Achsen keine mechanische Kopplung und es kann eine Aufteilung in eine Leitachse und mehrere Folgeachsen vorgenommen werden. Beispiele sind elektronische Getriebe, elektronische Kurvenscheiben etc.

Diese Objekte können über die in JetSym ST/STX integrierten Funktionen (MotionControl-Befehle und -Setup) angesprochen werden, wenn sie **aktiv** sind. Achsverbunde müssen zuerst mithilfe des MotionGroup-Befehls explizit aktiviert werden. Einzelachsen sind implizit aktiv, wenn sie an keinem aktiven Achsverbund beteiligt sind. Wurde also ein Achsverbund aktiviert, so können die beteiligten Achsen nicht mehr einzeln angewiesen werden.

Folgende Grafik veranschaulicht, wie Achsen und Achsverbunde vor und nach ihrer Aktivierung angesprochen werden können:



Anmerkung: Bei einem inaktiven Objekt sind nur die Befehle zur Aktivierung und zur Diagnose erlaubt. Der Befehl zum Rücksetzen der Fehler ist auch bei einem inaktiven Objekt zulässig, er muss aber nicht notwendigerweise abgesetzt werden.

Durch die Betrachtung von Achsverbunden als eigenständige Objekte wird die Programmierung von koordinierten Bewegungen erleichtert und systematisiert. Die Programmierung eines Bahnverbundes erfolgt z.B. unabhängig von der tatsächlichen Achskonfiguration in einem raumfesten Anwenderkoordinatensystem. Wird z.B. ein Technologieverbund mit einer entsprechenden Funktion gestoppt, halten automatisch alle beteiligten Achsen an.

Sofern es technologisch erforderlich ist, dass Einzelachsen innerhalb eines Verbunds ein gesondertes Verhalten aufweisen (z.B. Auskoppeln einer fliegenden Säge, Ändern von Reglerparametern während der Bewegung), so wird innerhalb des Achsverbundes dafür eine Funktion zur Verfügung gestellt.

Im **zustandsorientierten Objektmodell** der MCX-Software befinden sich die aktiven Objekte immer in einem definierten Betriebszustand (siehe auch unter [Zustandsverwaltung](#)). Abhängig von diesem Betriebszustand kann über mehrere Betriebskanäle die Programmierung, Parametrierung und Diagnose erfolgen.

Der **Nutzen der Zustandsorientierung** ist folgender:

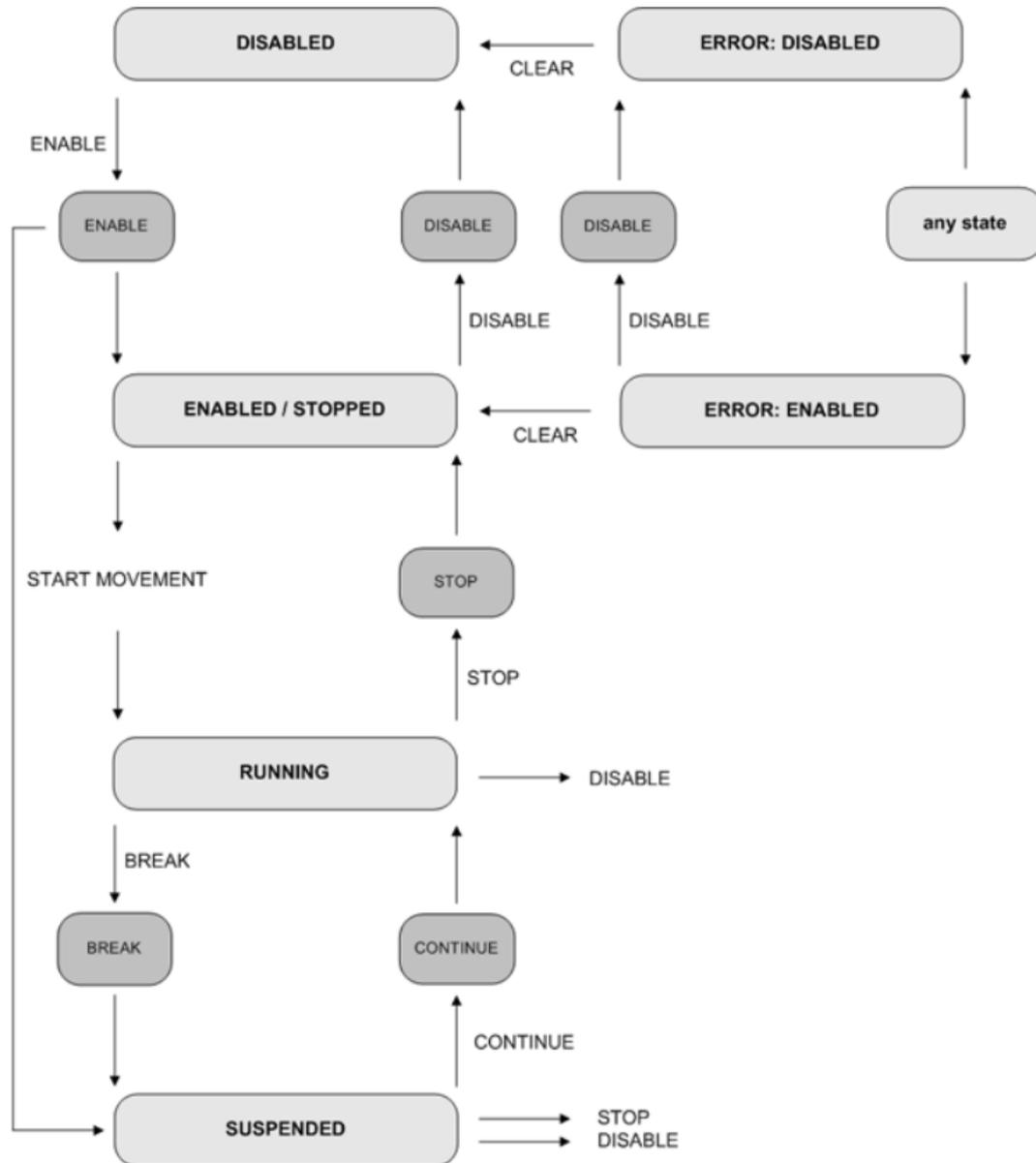
- Unsinnige Funktionsaufrufe oder Programmierfehler werden vom System erkannt und nicht angenommen
- Dem Anwender werden unzulässige Funktionsaufrufe gemeldet
- Das System hat eine wesentlich geringere Fehleranfälligkeit
- Unterstützung der Programmierung und Achs-/Achsverbund-Inbetriebnahme durch strukturiertes Vorgehen

1.2 Zustandsverwaltung

In der Regel werden die Objekte der MotionControl-Software (Achsens/Achsverbunde) direkt über die MotionControl-Befehle in JetSym ST/STX beauftragt, sodass Betriebszustandsübergänge dadurch automatisch ausgelöst werden

Zur direkten Zustandsverwaltung - z. B. zur Reaktion im Fehlerfall - ist in JetSym STX der MotionApi-Befehl `<Object>.State.Transitions` vorgesehen. Durch diesen Befehl sowie durch den Befehl `<Object>.Diagnostics.ClearErrors()` können explizite Betriebszustandsübergänge ausgelöst werden.

Die möglichen Betriebszustandsübergänge für ein Objekt sind in folgender Grafik veranschaulicht:



Anmerkungen:

Ein Objekt (Achsenverbunde erst nach Aktivierung) befindet sich immer nur in **einem** definierten Betriebszustand. Ist z. B. das Objekt im Betriebszustand *Läuft*, dann ist der Betriebszustand *Freigegeben* nicht aktiv (obwohl natürlich die Achsen freigegeben sind).

Die Kommandos der State.Transitions-Befehle werden auch im gepufferten Betrieb immer **sofort** wirksam. Sämtliche Vorlaufpuffer werden dabei geleert.

Im Einzelnen sind die Betriebszustände wie folgt charakterisiert:

GESPERRT (State.IsDisabled):

Achse: Die Regelkreise des Achs-Antriebs sind geöffnet.

Technologie-/Bahnverbund: Die Regelkreise aller zum Verbund gehörenden Antriebe sind geöffnet.

FREIGEGEBEN/ANGEHALTEN (State.IsEnabled):

Achse: Die Regelkreise des Achsantriebs sind geschlossen. Der Achsinterpolator steht. Sämtliche Vorlaufpuffer des Funktionskanals sind geleert.

Technologieverbund: Die Regelkreise aller zum Verbund gehörenden Antriebe sind geschlossen. Die Achsinterpolatoren sämtlicher zum Verbund gehörenden Achsen stehen.

Bahnverbund: Die Regelkreise aller zum Verbund gehörenden Antriebe sind geschlossen. Der Bahninterpolator steht. Sämtliche Vorlaufpuffer des Funktionskanals sind geleert.

LÄUFT (State.IsRunning):

Achse: Der Sollwertgenerator läuft. In der gepufferten Betriebsart werden die Vorlaufpuffer des Funktionskanals abgearbeitet und können auch weiterhin gefüllt werden.

Technologieverbund: Mindestens der Achsinterpolator einer zum Verbund gehörenden Achse läuft.

Bahnverbund: Der Bahninterpolator läuft. Die Vorlaufpuffer des Funktionskanals werden abgearbeitet und können auch weiterhin gefüllt werden.

UNTERBROCHEN (State.IsSuspended):

Dieser Betriebszustand ist der gleiche wie **FREIGEGEBEN** mit der Ausnahme, dass sämtliche Vorlaufpuffer gefüllt bleiben, aber gesperrt sind. Eine unterbrochene Bewegung kann über den State.Transitions-Befehl „Resume“ (Zustandsübergang *Wiederaufnehmen*) wiederaufgenommen werden.

FEHLER: GESPERRT/FEHLER: FREIGEGEBEN:

Fehlerzustände, die nach internen Fehlerreaktionen erreicht werden.

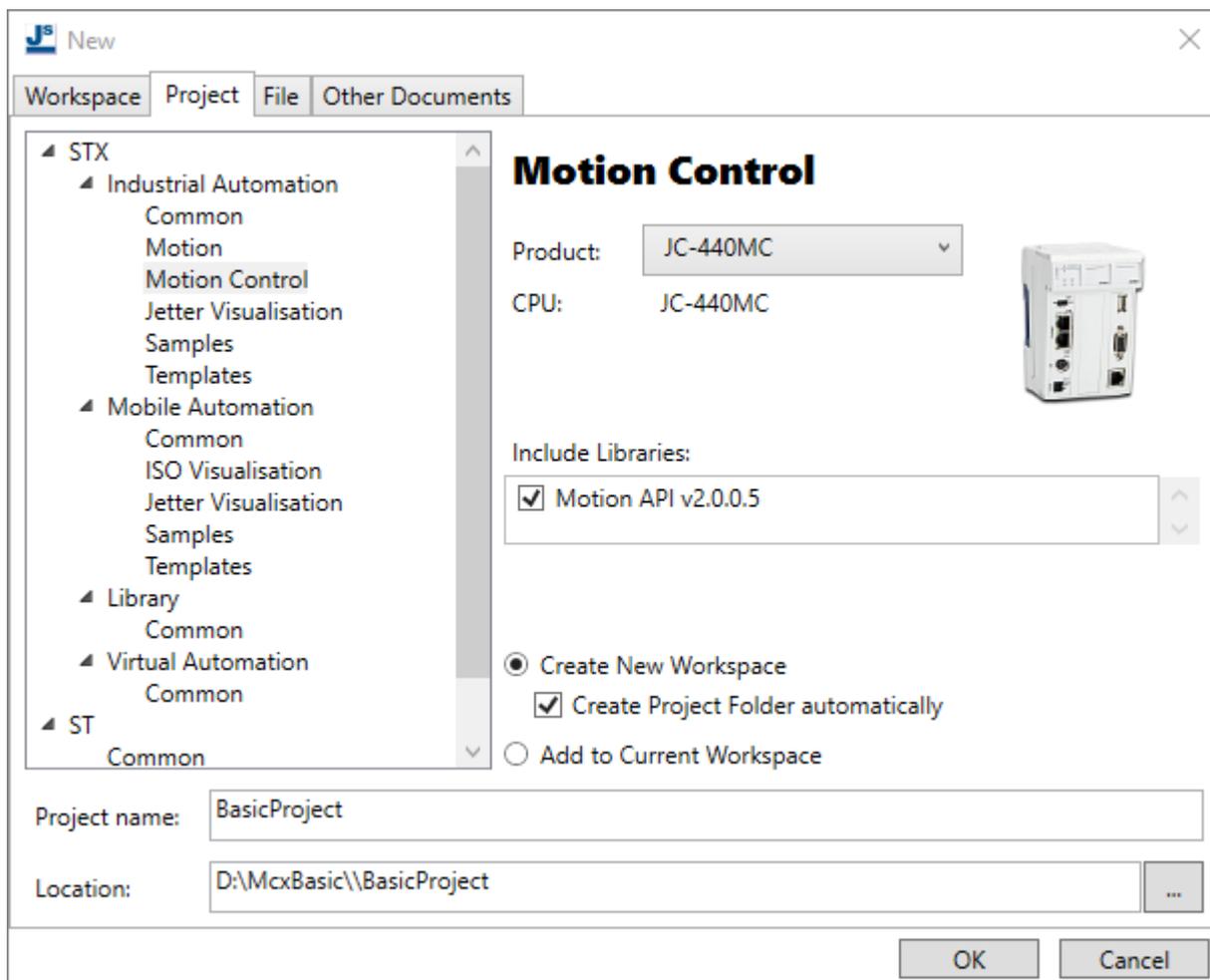
2 Konfiguration der MCX-Steuerung

Voraussetzungen:

- JetSym 5.6.1
- Maus mit Rechtshänder-Konfiguration:
 - o Untermenü öffnen: Rechter Mausklick
 - o Doppelklick: Zweimaliges schnelles Klicken mit der linken Maustaste
 - o Auswahl: Einmaliger Klick auf die linke Maustaste

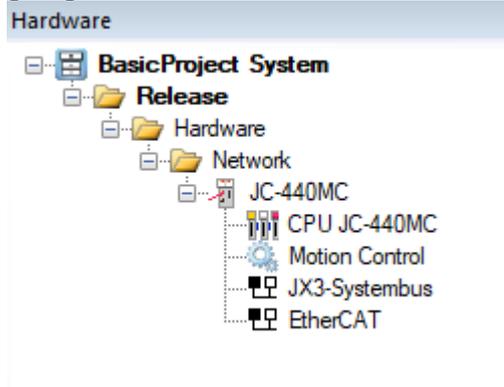
2.1 Anlegen eines MCX-Projekts

In JetSym über „Datei->Neu“ den Projekt-Wizard öffnen.



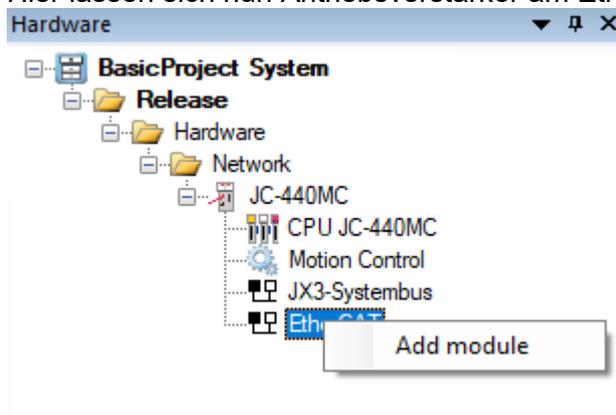
Wählen Sie eine MCX-Steuerung aus und binden Sie die MotionAPI ein. Es wird immer die aktuell installierte und zur Steuerung passende Version vorgeschlagen. Durch Bestätigung wird nun der Projektbaum erzeugt.

Im Hardwaremanager ist nun die Steuerung und die Basis-Knoten zur weiteren Konfiguration angelegt.

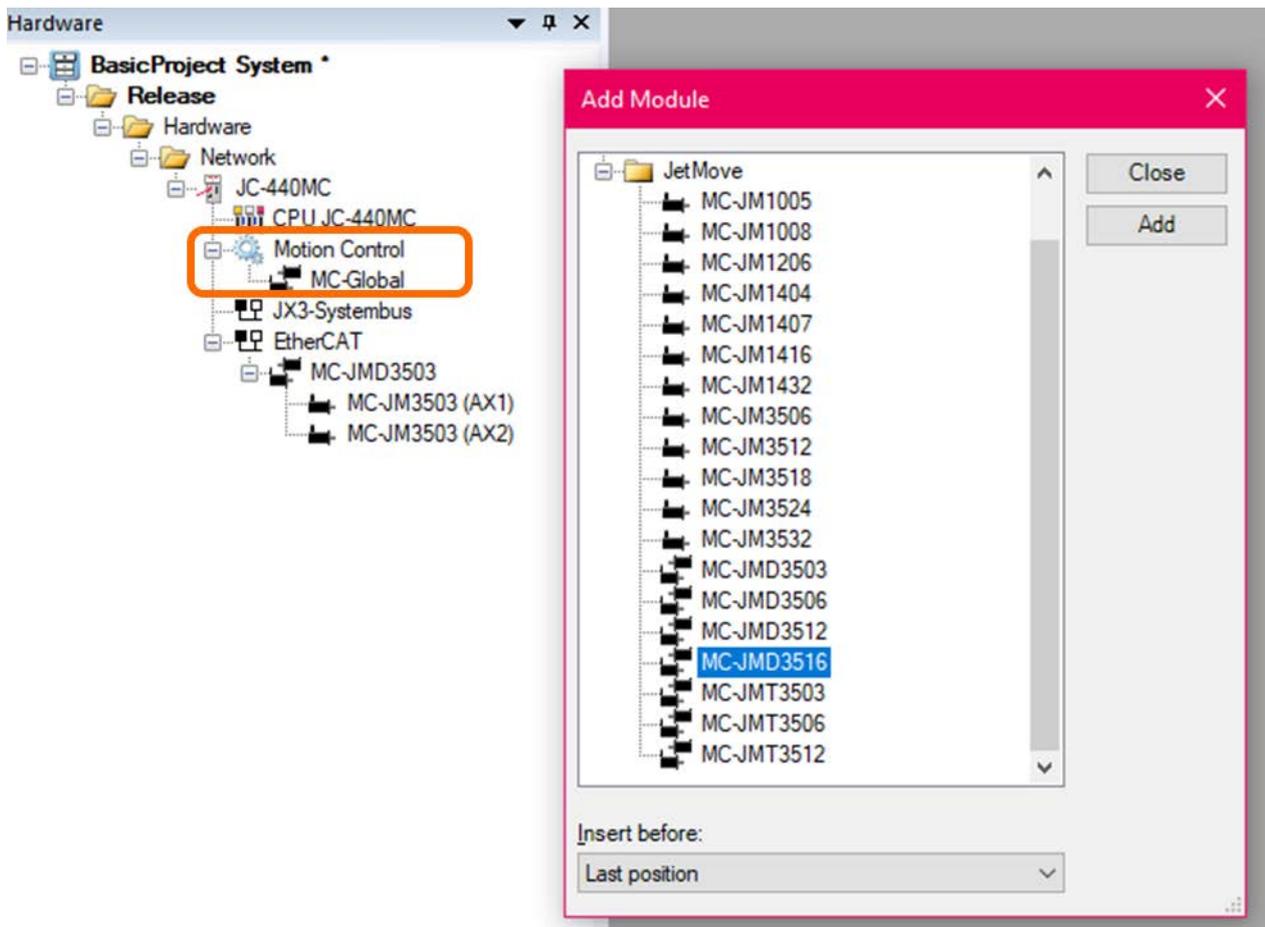


2.1.1 Antriebsverstärker hinzufügen

Hier lassen sich nun Antriebsverstärker am EtherCAT anlegen.



Über das Untermenü am Knoten „EtherCAT“ öffnet sich das Dialogfeld zur Auswahl der Module, die per "Add module" an diesen Knoten angeschlossen werden können.

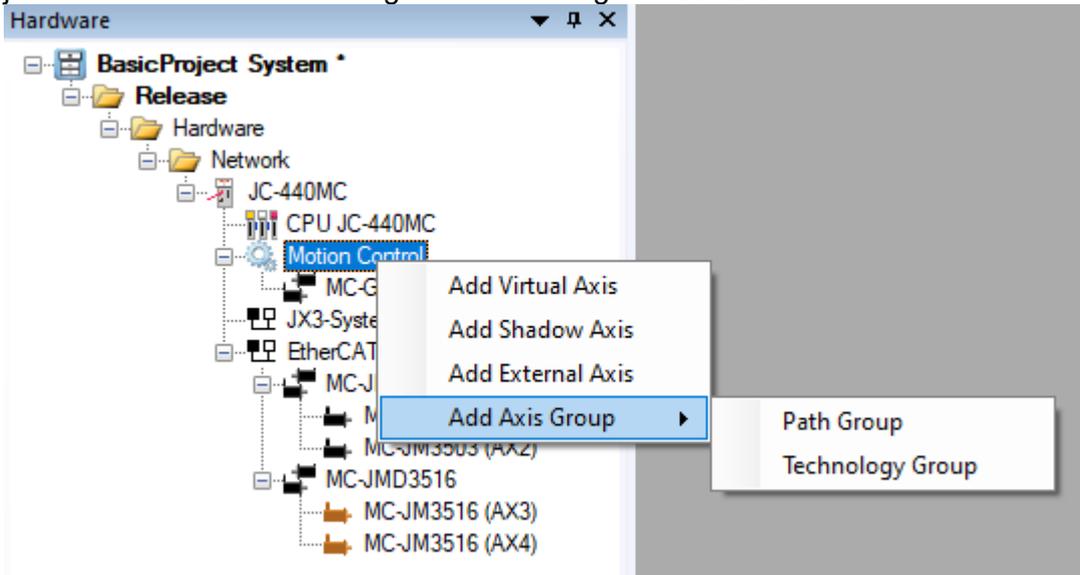


Wählen Sie nun die benötigten Antriebsverstärker aus und fügen Sie diesen per Doppelklick oder „Hinzufügen“ (Add) der Hardwarekonfiguration hinzu.

Mit dem Anlegen einer MC-Achse wird nun auch der Knoten „MC-Global“ angelegt.

2.1.2 Weitere MC-Objekte anlegen

Öffnen Sie das Untermenü auf „Motion Control“. Sie können nun auswählen, welche anderen Objekte Sie Ihrer Hardwarekonfiguration hinzufügen möchten.



Virtuelle Achse: Diese Achse besitzt keine eigene Hardware, die Sollwerte werden aber wie bei einer realen Achse berechnet.

Schattenachse: Diese Achse erbt die Soll- und Istwerte von einem angelegten Achsobjekt.

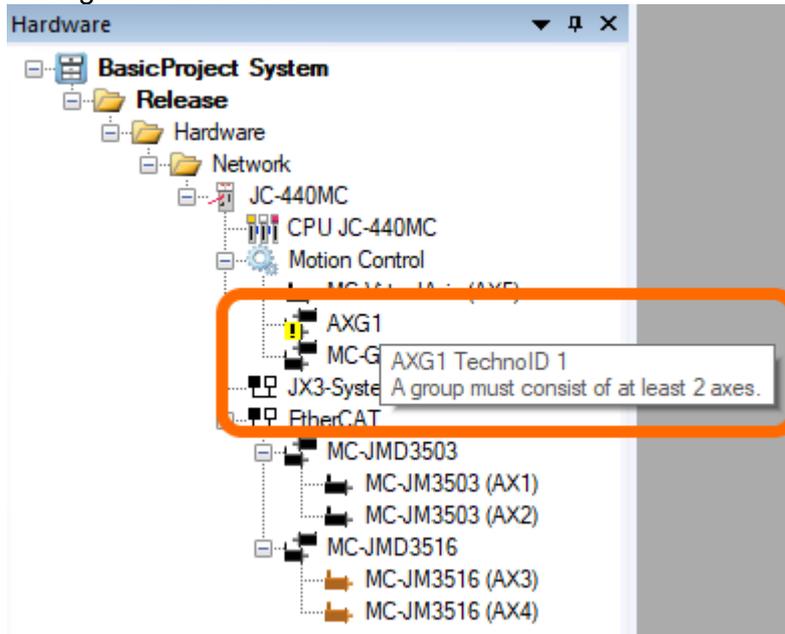
Externe Achse: Hier kommen die Istwerte von einem Zählermodul wie z.B. dem zweiten Gebereingang bei einem JM-3000. Dadurch lassen sich z.B. Reibräder zur Positionserfassung anschließen. Die Achse kann nicht aktiv von der MC gesteuert werden.

Bahnverbund: Anlegen eines Bahnverbunds für kartesische oder SCARA-Kinematiken.

Technologieverbund: Anbinden von Folgeachsen an eine Leitachse per elektrischem Getriebe oder Kurvenscheibe.

2.1.3 Inkorrekte Hardwarekonfiguration

Sind Unstimmigkeiten in der Hardwarekonfiguration vorhanden, wird dies durch ein Ausrufezeichen gekennzeichnet.

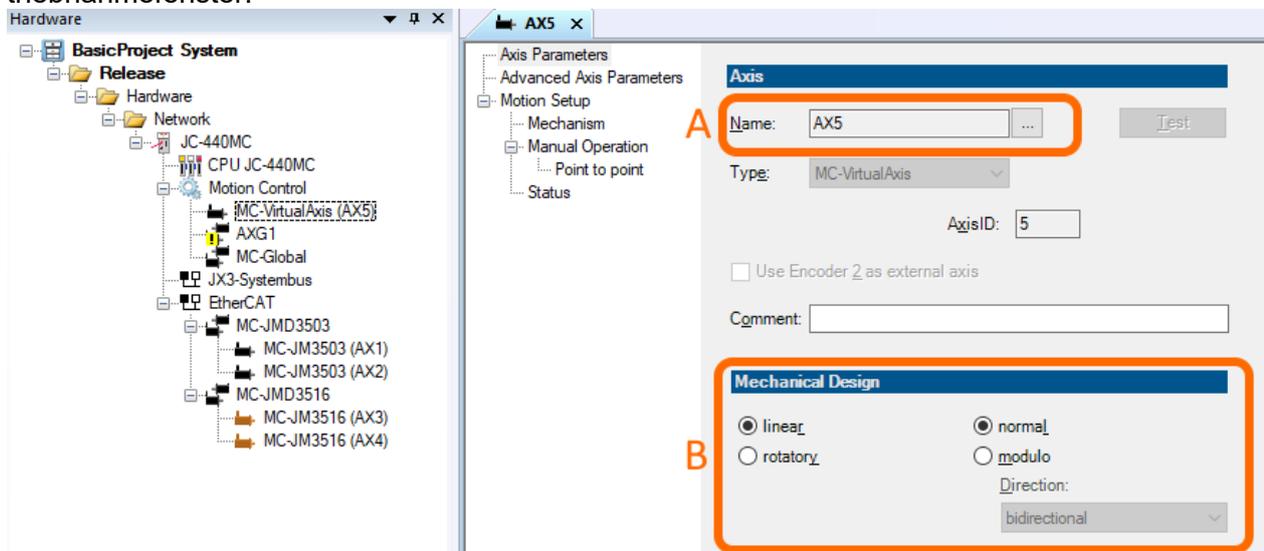


In diesem Beispiel ist der Technologieverbund „AXG1“ angelegt, es fehlt aber noch die Konfiguration, welche Achsen dem Verbund zugewiesen sind.

Wenn sie mit dem Cursor über das markierte Objekt fahren, zeigt JetSymb eine kurze Erklärung.

2.1.4 Konfiguration Achsobjekte

Per Doppelklick auf ein Achsobjekt öffnet sich der MotionSetup – ein Konfigurations- und Inbetriebnahmefenster.



In diesem Beispiel ist das MotionSetup der virtuellen Achse „AX5“ geöffnet.

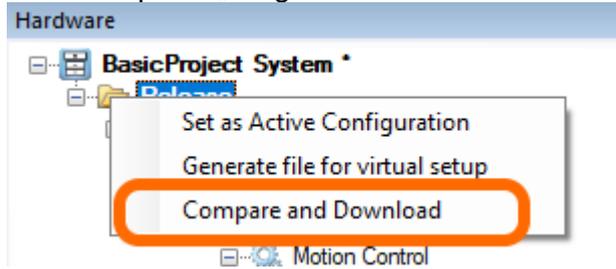
Auf der Startseite des MotionSetups lassen sich die grundsätzlichen Einstellungen des Achsobjekts festlegen.

- A. Es empfiehlt sich, statt des vorgeschlagenen Achsnamens „AX5“ einen für die Anwendung sinnvollen Namen zu vergeben. Über diesen Namen wird das jeweilige Achsobjekt auch im STX-Programm angesprochen.
- B. Bei der „Mechanischen Ausführung“ wird festgelegt, ob wir hier eine rotatorische oder lineare Bewegung haben und ob die Achse eine Modulo-Achse oder normale Achse ist.

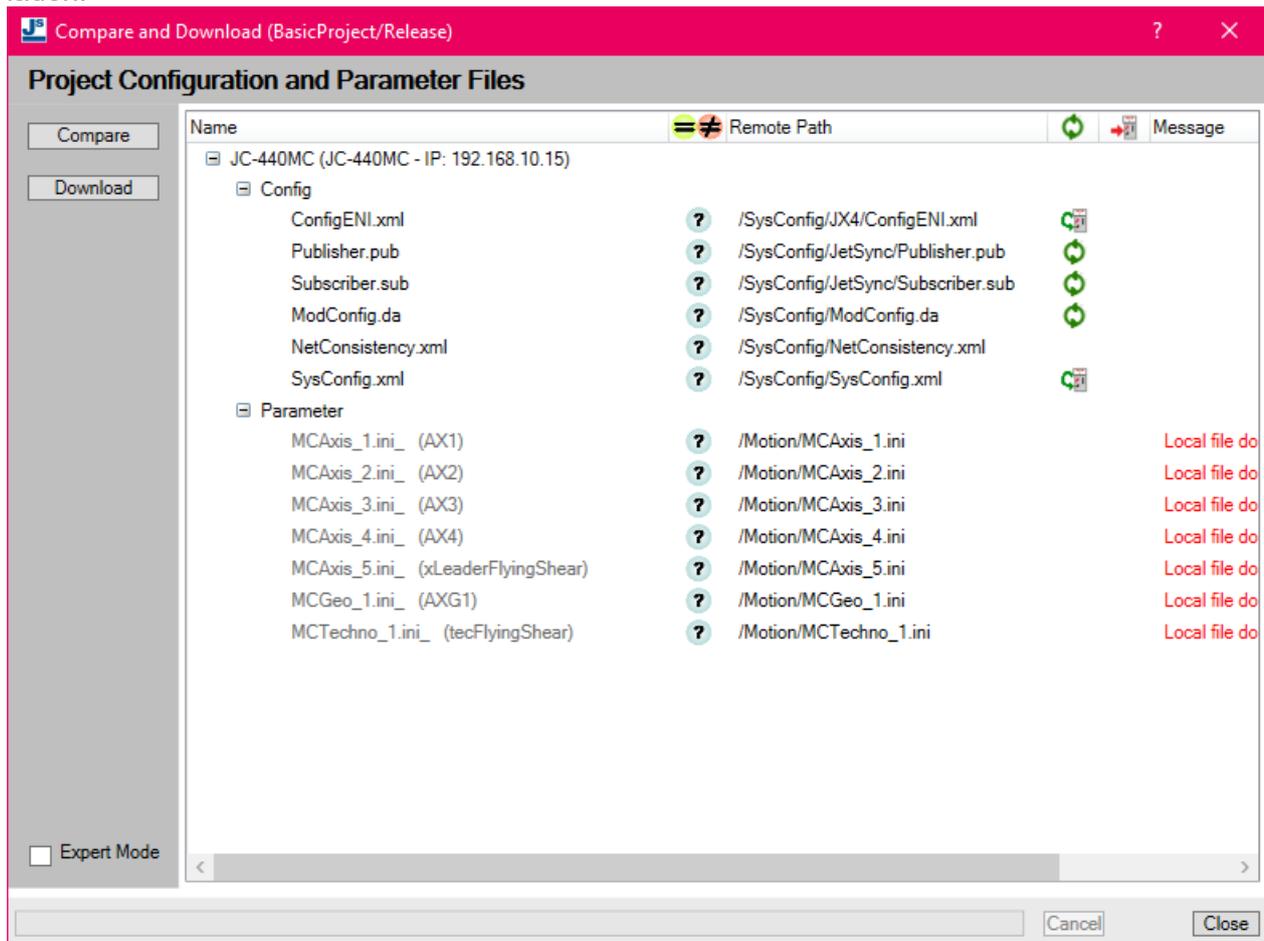
- Normale Achse: Die Achse hat einen festen Verfahrbereich, der nicht verlassen werden kann.
 - Modulo Achse: Die Position der Achse springt beim Überschreiten des Modulobereichs wieder auf Anfang.
- Es wird hierbei meist die Bewegung am Arbeitspunkt berücksichtigt.

2.1.6 Konfiguration übertragen

Im Untermenü auf dem Knoten der jeweiligen JetSym-Konfiguration – hier „Release“ – wählen Sie den Menüpunkt „Vergleichen und Downloaden“.



Es öffnet sich das Dialogfeld, um die Dateien für die Hardwarekonfiguration auf die Steuerung zu laden.



Ist von einer Achsinbetriebnahme die Konfigurationsdatei für dieses Achsobjekt vorhanden, kann die Konfiguration auch hierüber auf die Steuerung übertragen werden.

Bei Auswahl von „Vergleichen“ werden die lokalen Dateien mit denen auf der Steuerung verglichen.

Mit Auswahl des Download-Buttons beginnt die Übertragung. Hat sich die Hardwarekonfiguration geändert, wird ein Neustart der Steuerung angeboten.

Erst durch einen Neustart wird eine neue Hardwarekonfiguration aktiv.

Im MotionSetup des MC-Global (MotionSetup -> Status) lässt sich nach dem Neustart überprüfen, ob der MCX-Kernel mit dieser Hardwarekonfiguration erfolgreich starten konnte.

The screenshot shows the MCX software interface. At the top, there is a navigation bar with buttons for 'Load', 'Save', 'Upload', 'Download', and 'Help'. Below this is a main window with a left sidebar containing 'Diagnose', 'Errors', 'Status', and 'Files'. The main area displays a log of messages with columns for 'Time', 'Axis / Axis group', and 'Message'. The messages indicate that four axis groups (tecGroup01-04) have had their default parameters set, the drives are synchronous, and the MC V. 1.21.2.7 software has been successfully initialized and started. There are also buttons for 'Start file-logging', 'Configuration', 'Open Log-File', and 'More messages'. At the bottom left, a status indicator shows 'No error.' with a globe icon.

Time	Axis / Axis group	Message
17:34:31.775	tecGroup01	Set default parameters
17:34:31.776	tecGroup02	Set default parameters
17:34:31.777	tecGroup03	Set default parameters
17:34:31.777	tecGroup04	Set default parameters
17:34:31.790	Setup	Drives synchronous
17:34:35.783	Setup	MC V. 1.21.2.7 successfully initialized and started

3 Programmierung der MCX

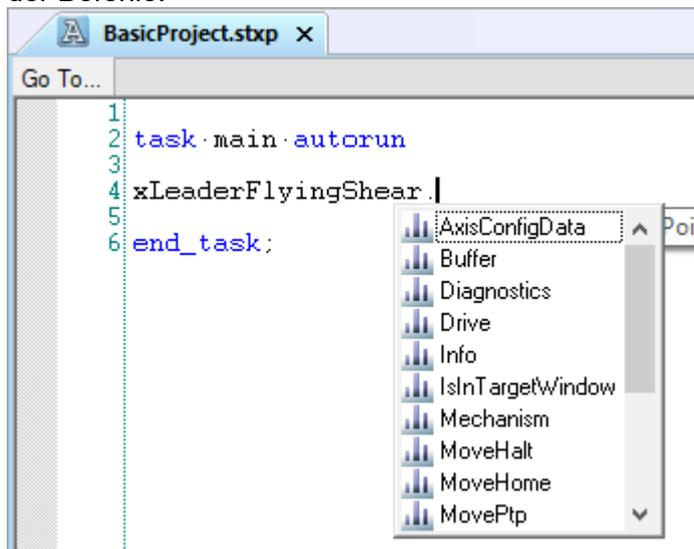
Die Programmierung der MCX in STX erfolgt über die MotionAPI. Dies ist eine objekt-orientierte Bibliothek, in der die verfügbaren Funktionen und Eigenschaften logisch gegliedert sind.

Anhand der Hardwarekonfiguration wird im Hintergrund die HardwareConfig.stxp erzeugt. Hierbei werden die MC-Objekte wie Achse, Verbunde aber auch ein MManager für die MotionAPI angelegt.

Damit die Objekte im Programm verfügbar sind, ist ein fehlerfreier Kompilervorgang nötig. Daher kann ein Default-Task hilfreich sein, um ohne Fehler kompilieren zu können.

```
task main autorun
end_task;
```

Da nun für die MotionAPI alle Objekte angelegt sind, unterstützt die IntelliSense bei der Eingabe der Befehle.



In der MotionAPI sind alle verfügbaren Funktionen des jeweiligen Objekts abgebildet.

3.1 Beispielprogramm

Hier ein kurzes Beispielprogramm, um zu zeigen, wie die Programmierung aussehen kann und welche Elemente vorhanden sind.

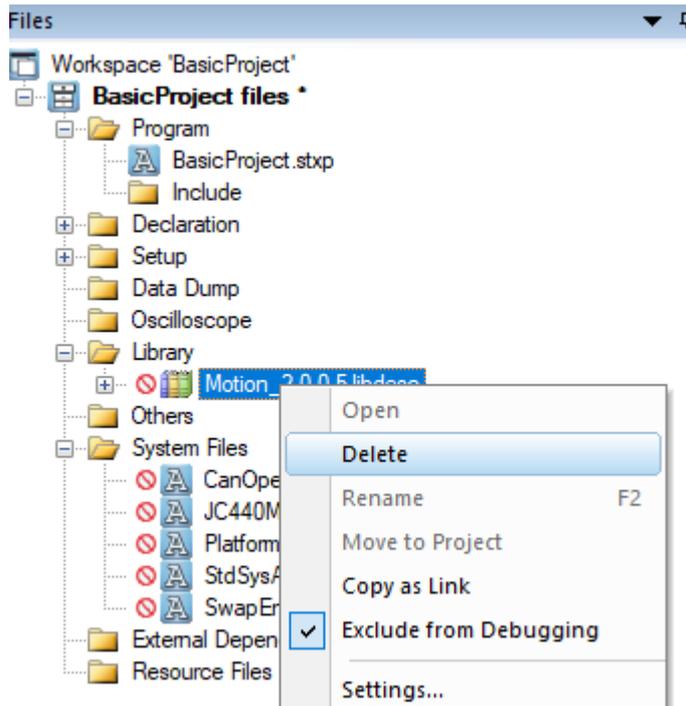
```
task main autorun
  xLeaderFlyingShear.MoveHome.SetReference(0.0);           // A
  xLeaderFlyingShear.Power.Enable();                       // A
  when xLeaderFlyingShear.State.IsEnabled continue;       // B
  tecFlyingShear.Activate();                               // C
  when tecFlyingShear.State.IsEnabled continue;           // B
  tecFlyingShear.MoveVelocity.Start(xLeaderFlyingShear);  // D
  when tecFlyingShear.State.IsRunning continue;          // B
  when tecFlyingShear.Mechanism.Slope.IsAtConstantSpeed continue; // B1
  tecFlyingShear.MoveHalt.Start(xLeaderFlyingShear, MCTechnoHaltModes.Normal); // D
  when tecFlyingShear.State.IsEnabled continue;           // B
  tecFlyingShear.Deactivate();                             // C
  when tecFlyingShear.State.IsInactive continue;         // B
  xLeaderFlyingShear.Power.Disable(MCPowerDisableModes.Forced); // A
  when xLeaderFlyingShear.State.IsDisabled continue;     // B
end_task;
```

- A. Einzelachskommandos, wie das Setzen der Grundstellung, Freigabe, Sperren
- B. Statusabfragen der MC-Objekte: Entweder Status der Einzelachse oder des Verbunds
B1: Abfragen von speziellen Status wie z.B. Rampenstatus
- C. Aktivierung und Deaktivierung eines Verbunds
- D. Verbundkommandos, wie das Positionieren und Anhalten von Verbundmitgliedern

3.2 MotionAPI-Version wählen

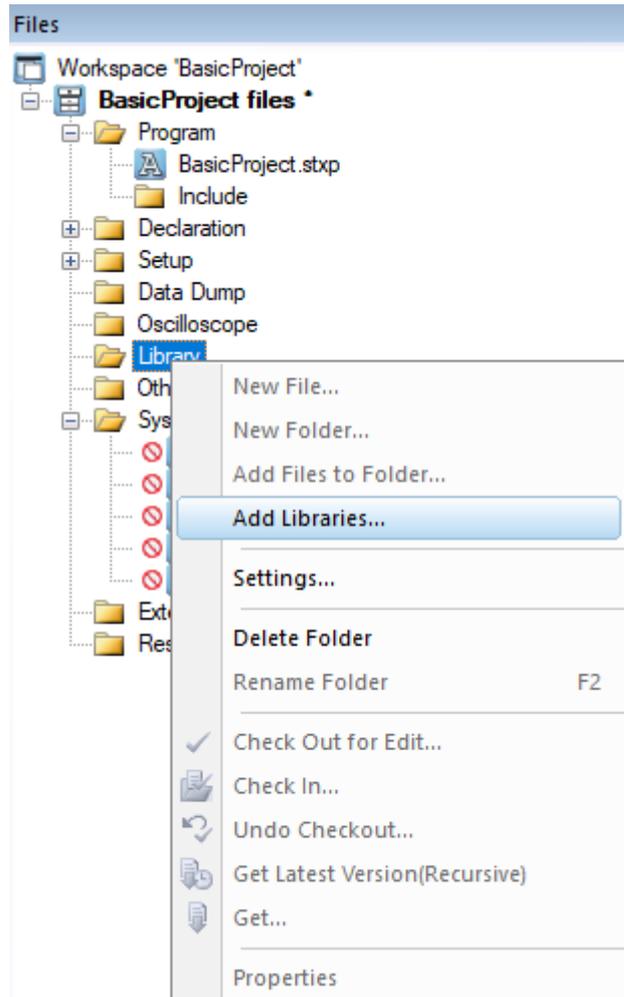
Beim Anlegen eines neuen Projekts wird immer die aktuelle, installierte Version vorgeschlagen. Soll aber eine bestimmte Version verwendet werden oder in einem bestehenden Projekt eine neuere Version eingebunden werden, dann sind folgende Schritte nötig.

1. Löschen der vorhandenen MotionAPI aus dem Projektbaum.

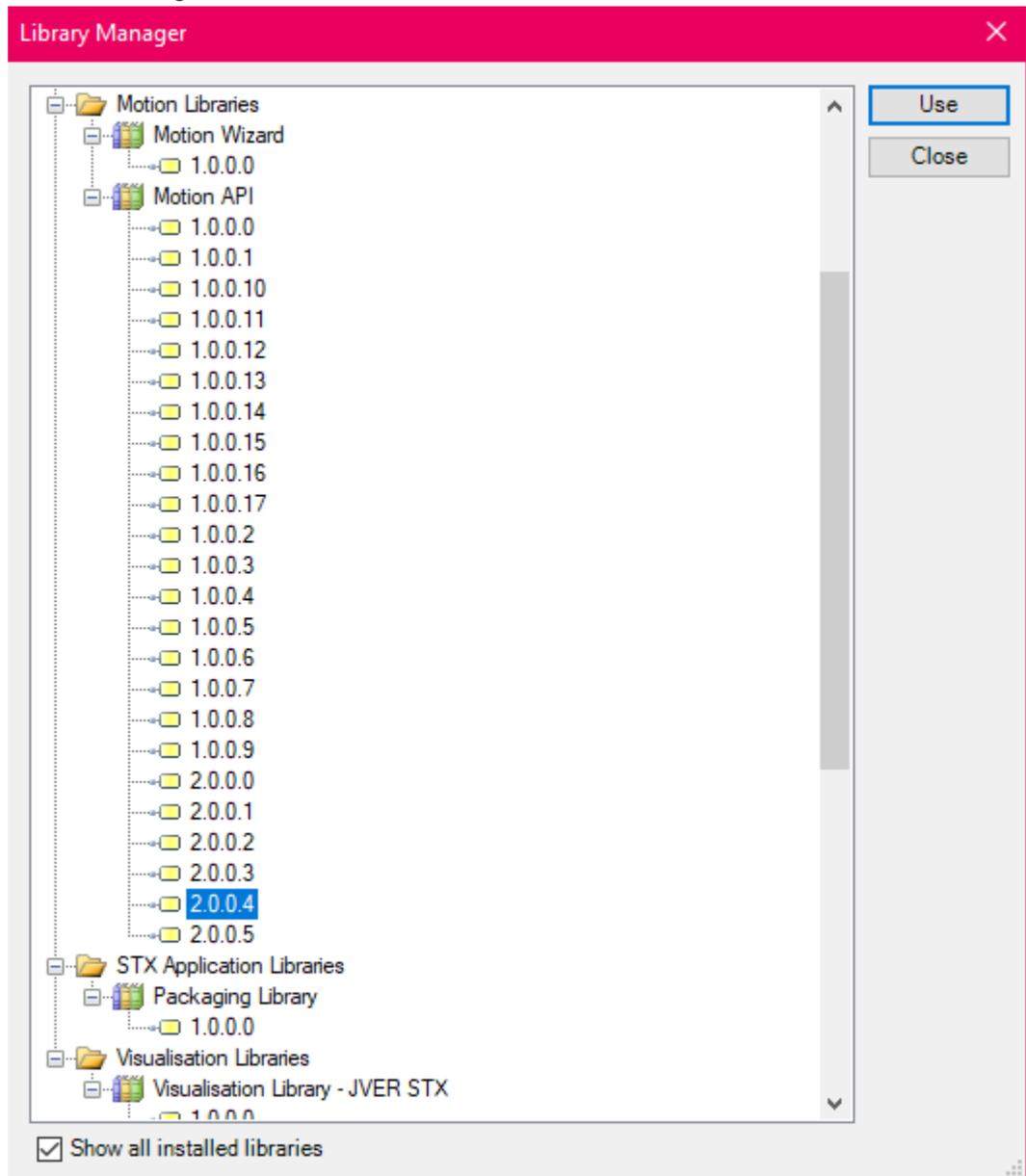


Es kann immer nur eine MotionAPI in einem Projekt eingebunden sein. Wird daher die eingebundene Version nicht entfernt, kann keine andere Version eingebunden werden.

2. Öffnen der Bibliotheksauswahl per Untermenü von „Bibliothek“ im Projektbaum



3. Auswahl der gewünschten MotionAPI-Version.



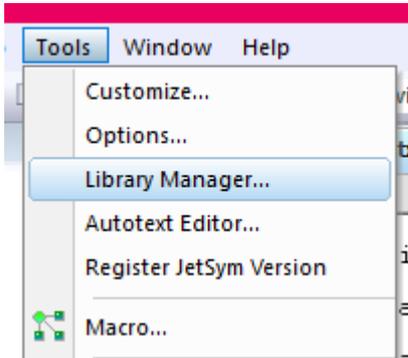
Ggf. muss „Alle installierten Bibliotheken anzeigen“ aktiviert werden.

Durch Klick auf „Verwenden“ wird nun die ausgewählte MotionAPI eingebunden.

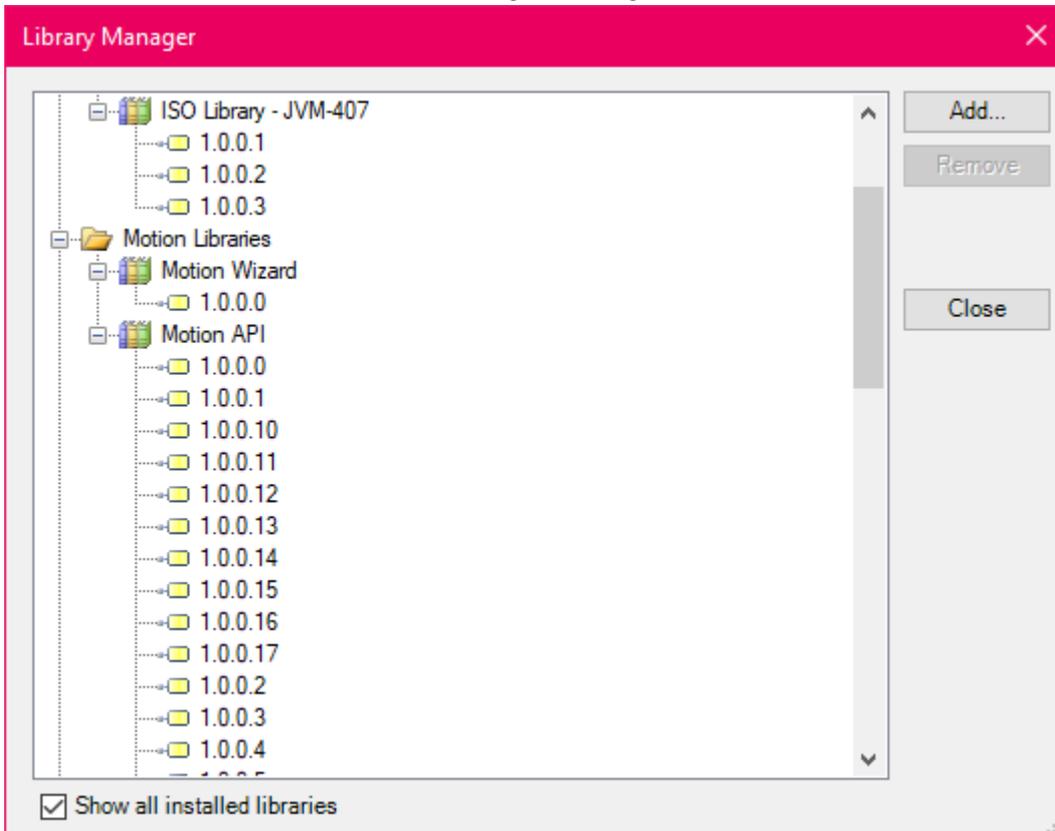
3.2.1 MotionAPI hinzufügen

Es können nur MotionAPI-Bibliotheken in einem Projekt eingebunden, die vorher der Bibliotheksverwaltung hinzugefügt wurden.

1. Öffnen der Bibliotheksverwaltung.



2. Es öffnet sich die Bibliotheksverwaltung und zeigt die installierten Bibliotheken an.



3. Per „Hinzufügen“ öffnet sich der „Datei öffnen“-Dialog. Wählen sie das „libpackage“ aus, welches hinzugefügt werden soll.
4. Ggf. muss die Option „Alle installierten Bibliotheken anzeigen“ aktiviert werden, damit die Bibliotheken angezeigt werden.
5. Nun können Sie wie unter „Auswahl MotionAPI“ beschrieben, die zusätzliche Bibliothek in Ihr Projekt einbinden.

Jetter AG
Gräterstraße 2
71642 Ludwigsburg
www.jetter.de

E-Mail info@jetter.de
Telefon +49 7141 2550-0