

# CPX-CMAX ADD-ON INSTRUCTIONS USER GUIDE

**EXTERNAL CONTROLLER:** Allen Bradley 1756-L61 PLC with 1756-ENBT/A EthernetIP Module  
**FESTO CONTROLLER:** CPX-CMAX  
**PROTOCOL:** FHPP (Festo Handling Positioning Profile) for the CMAX axis controller  
**CONTROL METHOD:** CPX-FB32 (EthernetIP)  
**DOCUMENT DATE OF ISSUE:** AUGUST 22/2014  
**DOCUMENT REVISION:** V2.0

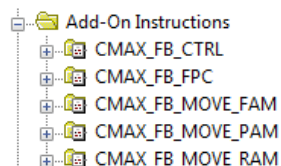
## Service

Please contact your local Festo service partner if you have any technical problems.

[www.Festo.com](http://www.Festo.com)

## Application Description:

This document describes add-on instructions (AOI) created for the Festo CPX-CMAX Servo Pneumatic Controller to be used with the Rockwell Automation RSLogix 5000 Software. With these instructions, axis controllers of type CPX-CMAX-C1-1 from Festo can be actuated/controlled.



## Attachments/Dependencies

- Rockwell Software RS5000 project file:
  - AB\_FB32\_CMAX\_V17\_AOI\_Customer\_20140822.ACD
  - AB\_FB32\_CMAX\_V18\_AOI\_Customer\_20140822.ACD
  - AB\_FB32\_CMAX\_V19\_AOI\_Customer\_20140822.ACD
  - AB\_FB32\_CMAX\_V20\_AOI\_Customer\_20140822.ACD
  - AB\_FB32\_CMAX\_V21\_AOI\_Customer\_20140822.ACD
- Rockwell Software RS5000 Data Types (More are within the PLC Project files –These are minimum requirements):
  - ud\_CMAX\_Control.L5X
  - ud\_CMAX\_Modes.L5X
- Rockwell Software RS5000 Add-On Instructions:
  - CMAX\_FB\_CTRL.L5X
  - CMAX\_FB\_FPC.L5X
  - CMAX\_FB\_MOVE\_FAM.L5X
  - CMAX\_FB\_MOVE\_PAM.L5X
  - CMAX\_FB\_MOVE\_RAM.L5X
- Festo FCT (Festo Configuration Tool Software) project file:
  - DemoRm Bench CPX-CMAX With DGCI via FB32.V1.0.0.zip
  - DemoRm Bench CPX-CMAX With DSMI via FB32.V1.0.0.zip
- Festo FMT (Festo Maintenance Tool Software) project file:
  - AB\_to\_FB32\_to\_CMAX\_FHPP.cpx
- Festo Manual(Additional information for operating and setting up the CPX-CMAX controller):
  - CPX-CMAX FHPP for Control 559757g1 0908NH.pdf
- Revision History:
  - CPX-CMAX to AB via FHPP EthernetIP AOI Revision History ca0pro.docx

## General Information

Your Obligations With Respect to Using the attached Content:

Your right and ability to use the code contents are limited, subject to the terms and conditions as stated herein.

All use of available content & Attachments will be at your sole risk.

You acknowledge that Festo has not reviewed or evaluated (nor is obligated to do so) the materials attached and expressly agree that Festo will have no obligation or liability of any kind with respect to the use or utility of any such materials by you or others.

You will not use the attached content for any purpose or in any manner that is unlawful or that violates any intellectual property right, privacy right, or other right of Festo or any third party.

The copyright in the material attached is held by Steven Proud.

You may not delete or alter or attempt to delete or alter any attribution, legal notice, trademark or copyright notice appearing on any attached content and such notices or marks must be included on any copy you make.

Furthermore, you will not reproduce, redistribute or repost on other web sites materials from the attached unless expressly authorized to do so by Steven Proud. You must take your own precautions to ensure that whatever content you have been supplied from the attachments is free of computer viruses or Trojan horses that may interfere with or damage the functionality of the operations of your or any third party computer system.

This version of software/firmware is a test version.

Festo takes no warranty for the functionality of the software or consequential damage by using this software.

This software is to be used only for evaluation purposes and it is not allowed to be given away to third parties.

### Service

Please contact your local Festo service partner if you have any technical problems.

[www.Festo.com](http://www.Festo.com)

## General Information

As with all AOI (Add-On Instruction), the user can import these themselves directly into a new project.

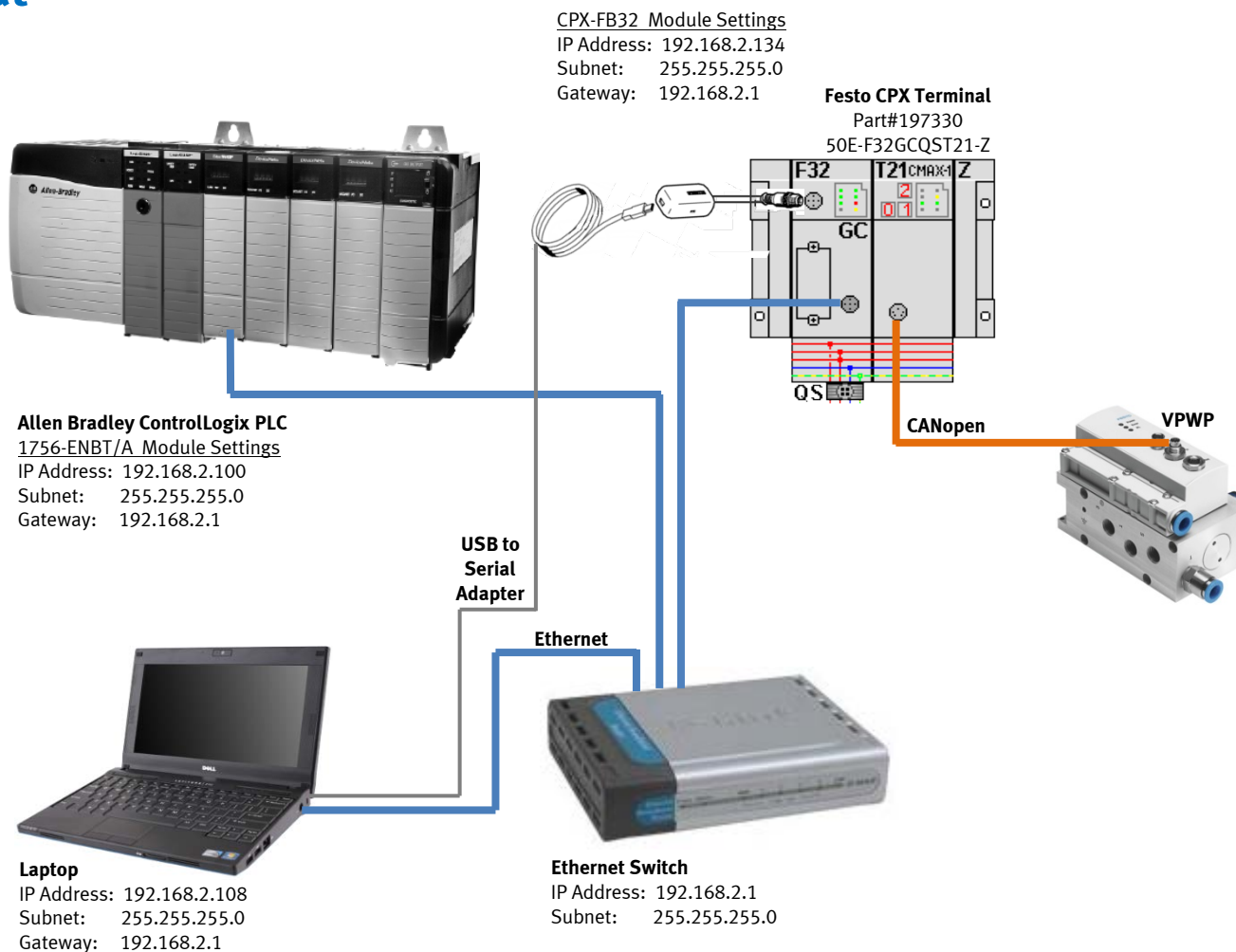
The AOI (Add-On Instruction) included with this application is no different, HOWEVER, with regards to the sample projects, all provided code is meant to be used in its **entirety**.

If you only need certain AOI (Add On Instruction), the code is designed so that the entire rung needs to be used.

Do **NOT** copy/use a partial amount of the rung, otherwise the function will be lost.

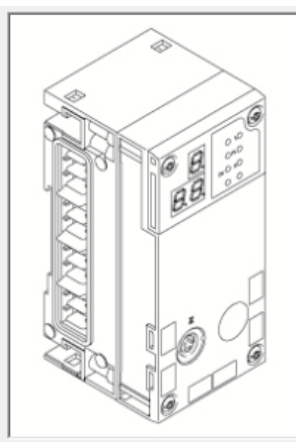
**NOTE:** Just as the AOI(Add On Instruction) samples are provided, each AOI (Add On Instruction) itself needs to be on a rung that has power at all times or it will not function correct. This is because the code inside the AOI (Add On Instruction) resets certain IO (Inputs/Outputs) when other bits are off. The only instruction that should be used prior to ANY of the AOI's are a bit that monitors the active communications between the PLC and the CPX Fieldbus module where the CPX-CPX-CMAX(Servo Pneumatic Controller) is installed.

## System Layout



# Servo Controller Information

Controller:		CPX-CMAX-C1-1
Modul position no.:		1
Device data	Compatibility	CPX-Node
Manufacturer device name:	CPX-CMAX-C1-1	
Manufacturer:	Festo AG & Co. KG	
Revision:	4	
Firmware version:	1.01.0808	
Build date:	07.01.2010 14:06:29	
FHPP version:	1.10	
Serial number:	99865894	
Order number:	548932	



**SERVO CONTROLLER:** **CPX-CMAX Controller**  
**SERVO CONTROLLER FW:** **V1.01.0808**  
**FCT VERSION:** **V1.2.1.3**  
**FCT PLUGIN VERSION:** **V1.1.0.92**  
**CONTROL METHOD:** **FHPP (Festo Handling Positioning Profile) for the CMAX axis controller**

## Allen Bradley Information

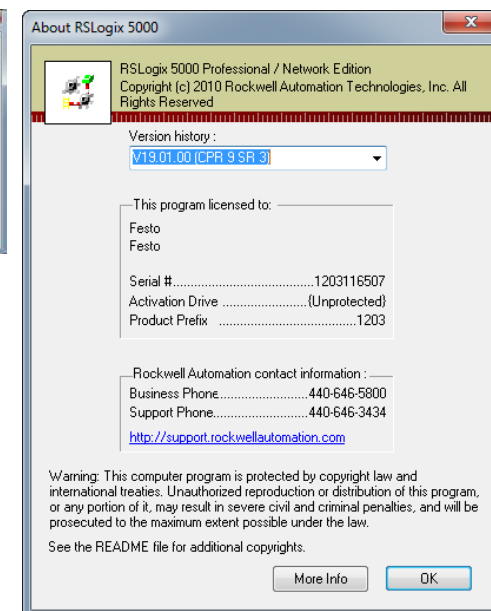
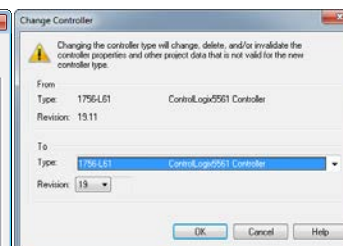
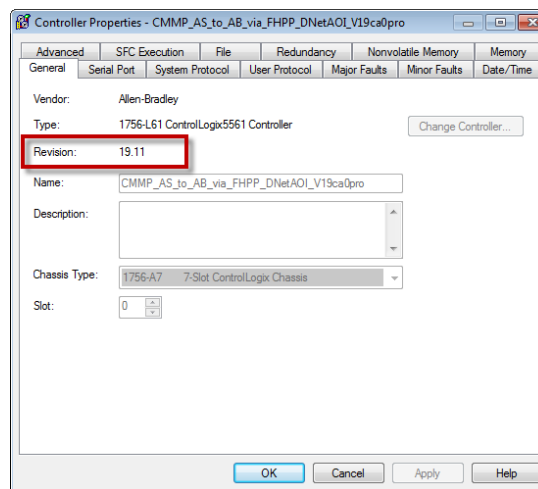


# Allen Bradley 1756-L61 Controller Information

## Features:

- ☐ 2Mb User Memory
- ☐ Non-volatile user memory → Compact Flash
- ☐ 1 x RS-232 serial port → ASCII, DF1, full/half-duplex, DF1 radio modem, DH-485
- ☐ Programming languages → Relay ladder, Structured text, Function block, SFC
- ☐ Communication Options → EthernetIP, ControlNet, DeviceNet, Data Highway Plus, Remote I/O
- ☐ Network connections → 256 EtherNetIP, 128 TCP (EN2x)

**PLC(LOGIC CONTROLLER): Allen Bradley 1756-L61 ControlLogix5561**  
**PLC CONTROLLER FIRMWARE: Controller V19.11**  
**RSLOGIX5000 SOFTWARE VERSION: V19.01.00 (CPR 9 SR 3)**





# Allen Bradley 1756-ENBT/A Ethernet Card Information

This is just shown here so we can verify the controller information

Module Properties: Local1 (1756-ENBT 4.1)

General | Connection | Module Info | Internet Protocol | Port Configuration | RSNetWork

Type: 1756-ENBT 1756 10/100 Mbps Ethernet Bridge, Twisted Pair Media [Change Type...](#)

Vendor: Allen-Bradley

Parent: Local

Name: ENET\_IP

Description:

Ethernet Address

☐ Private Network: 192.168.1.

☒ IP Address: 192 . 168 . 2 . 100

☐ Host Name:

Slot: 1

Module Definition

Revision: 4.1 [Change...](#)

Electronic Keying: Compatible Module

Rack Connection: None

Time Sync Connection: None

Status: Running

[OK](#) [Cancel](#) [Apply](#) [Help](#)

Module Properties: Local1 (1756-ENBT 4.1)

General | Connection | Module Info | Internet Protocol | Port Configuration | RSNetWork

Identification

Vendor: Allen-Bradley

Product Type: Communications Adapter

Product Code: 1756-ENBT

Revision: 4.8

Serial Number: 005A9B41

Product Name: 1756-ENBT/A

Status

Major Fault: None

Minor Fault: None

Internal State: Run mode

Configured: No

Owned: No

Module Identity: Match

[Refresh](#) [Reset Module](#)

Status: Running

[OK](#) [Cancel](#) [Apply](#) [Help](#)

Module Properties: Local1 (1756-ENBT 4.1)

General | Connection | Module Info | Internet Protocol | Port Configuration | RSNetWork

Internet Protocol (IP) Settings

IP settings can be manually configured or can be automatically configured if the network supports this capability.

☒ Manually configure IP settings

☐ Obtain IP settings automatically using BOOTP

☐ Obtain IP settings automatically using DHCP

☐ IP settings set by switches on the module

IP Settings Configuration

Physical Module IP Address: 192 . 168 . 2 . 100

Subnet Mask: 255 . 255 . 255 . 0

Gateway Address: 192 . 168 . 2 . 1

Domain Name:

Host Name:

Primary DNS Server Address: 0 . 0 . 0 . 0

Secondary DNS Server Address: 0 . 0 . 0 . 0

[Refresh communication](#) [Set](#)

Status: Running

[OK](#) [Cancel](#) [Apply](#) [Help](#)

Module Properties: Local1 (1756-ENBT 4.1)

General | Connection | Module Info | Internet Protocol | Port Configuration | RSNetWork

Auto-Negotiate	Speed	Duplex	Port
Selected	Current	Selected	Current
<input checked="" type="checkbox"/>	100 Mbps	Full	...

[Refresh communication](#) [Set](#)

Status: Running

[OK](#) [Cancel](#) [Apply](#) [Help](#)

NOTE: As indicated previously, all provided code is meant to be used in its **entirety** - the code is designed so that the entire rung needs to be used. Do **NOT** copy/use a partial amount of the rung, otherwise the function will be lost. Just as the sample code shows, each AOI (Add On Instruction) itself needs to be on a rung that has power at all times or it will not function correct.

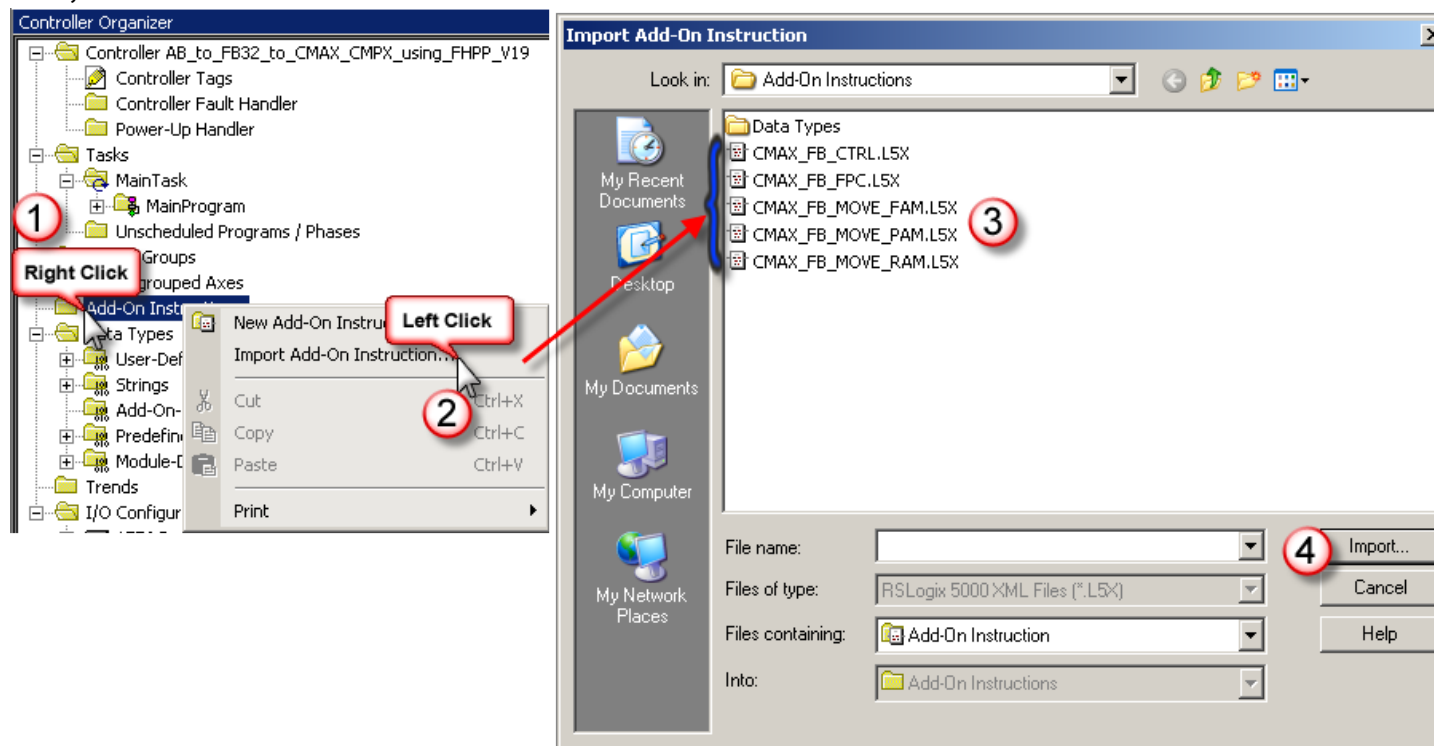
## Rockwell Software RS5000 Importing Add-On Instructions

This application package has come with a Rockwell Software RS5000 project file as indicated previously. All of the add-on instructions are already included in this file. In order to import these instructions and data types into another project, read the following:

There are 5 Rockwell Software RS5000 Add-On Instructions :

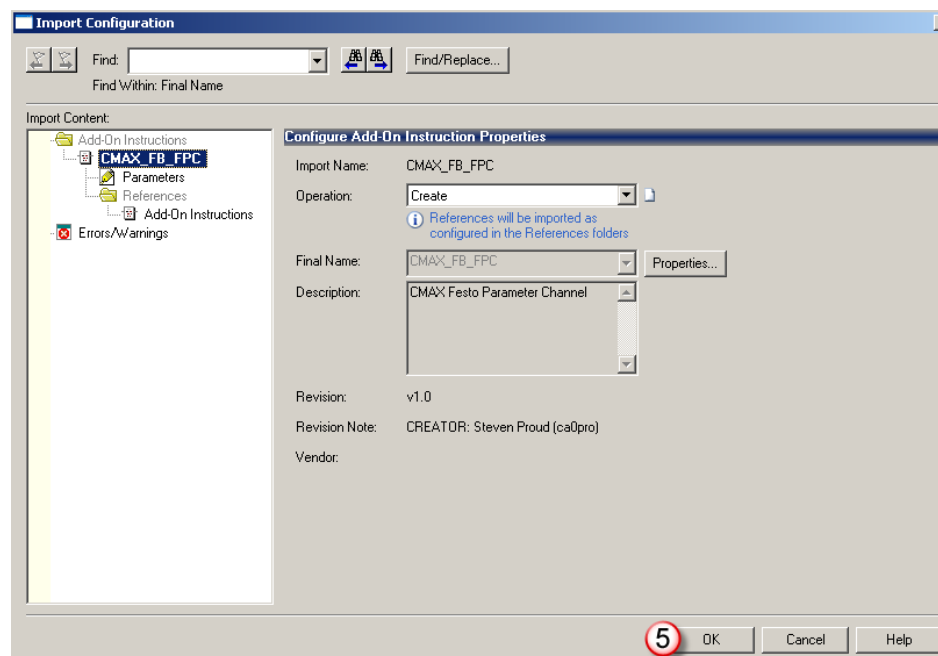
ud\_CMAX\_Control.L5X, ud\_CMAX\_Modes.L5X (referenced within the ud\_CMAX\_Control.L5X)

Import **all** of these as shown below within your project – the CMAX\_FB\_CTRL should be imported first, thereafter the order is NOT important as the software will import any referenced instructions.

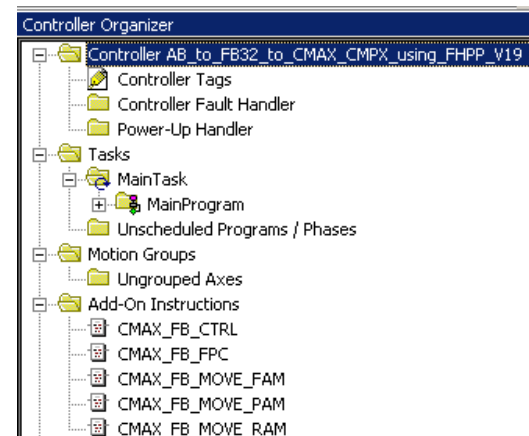


# Rockwell Software RS5000 Importing Add-On Instructions

The final step is to simply select the OK button.



When you have imported ALL instructions, your Add-On Instruction tree view should look like the following:



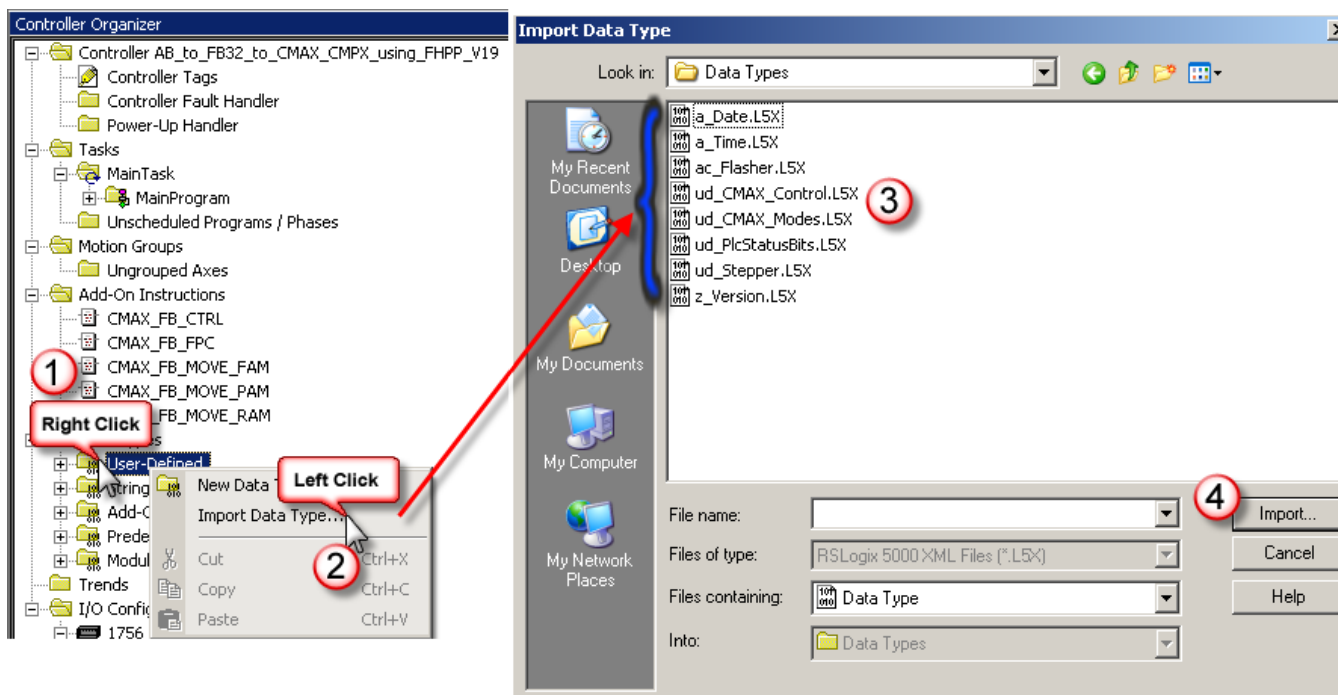
# Rockwell Software RS5000 Importing User-Defined Data Types

This application package has come with a Rockwell Software RS5000 project file as indicated previously. All of the data types are already included in this file. In order to import these instructions and data types into another project, read the following:

There are a minimum of 2 Rockwell Software RS5000 user-defined data types required :

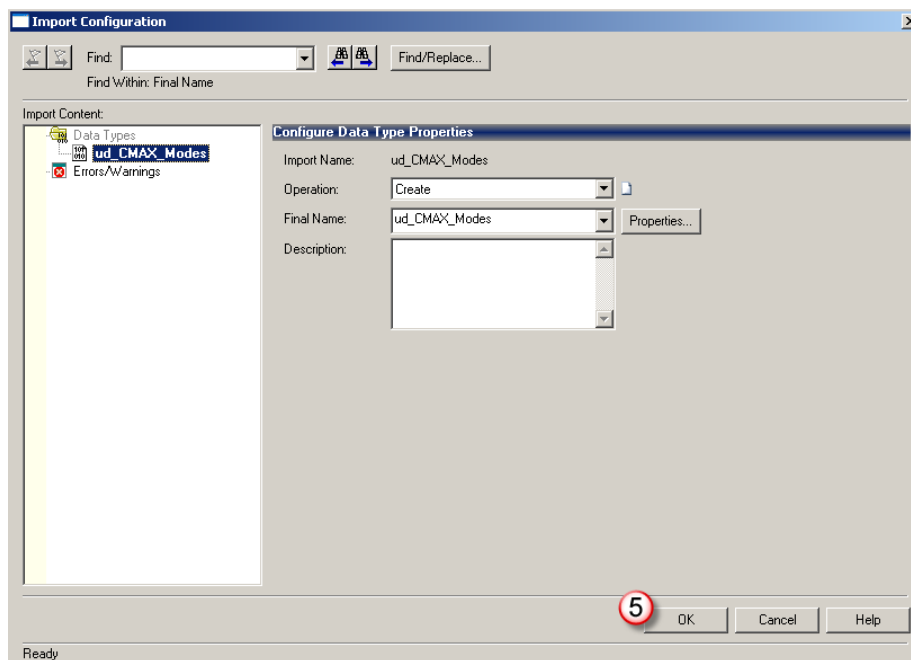
ud\_CMAX\_Control.L5X, ud\_CMAX\_Modes.L5X

Import either all the shown data types below or just the minimum 2 indicated above as shown below within your project – import the ud\_CMAX\_Modes.L5X (referenced within the ud\_CMAX\_Control.L5X) data type first and then the ud\_CMAX\_Control.L5X or you will encounter errors.

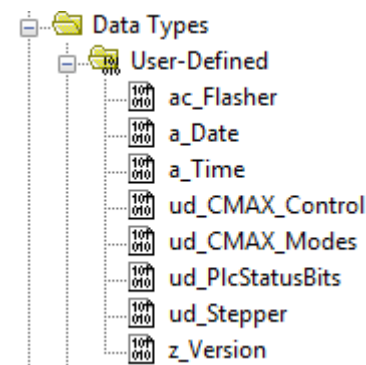


# Rockwell Software RS5000 Importing User-Defined Data Types

The final step is to simply select the OK button.



When you have imported ALL instructions, your Data Types tree view should look like the following:



# Rockwell Software RS5000 Data Types (User Defined)

For convenience and program organization, Data Type's have been provided. For each CPX-CMAX controller the user should create a tag linked to the data type "ud\_CMAX\_Control". In this sample, the tag created is "X\_Axis".

The screenshot displays the Rockwell RS5000 software interface, specifically the 'Controller Organizer' and 'Data Types' sections.

**Controller Organizer (Left Panel):** Shows the project structure for 'Controller AB\_FB32\_CMAX\_V19\_AOI\_Customer2012'. The 'Data Types' folder is expanded, showing a list of user-defined data types. A red arrow points to 'ud\_CMAX\_Control'.

**Data Types (Right Panel):** A list of user-defined data types is shown, including 'ac\_Flasher', 'a\_Date', 'a\_Time', 'ud\_CMAX\_Control', 'ud\_CMAX\_Modes', 'ud\_PlcStatusBits', 'ud\_Stepper', and 'z\_Version'. A red arrow points to 'ud\_CMAX\_Control'.

**Controller Organizer (Bottom Panel):** Shows the 'Controller Tags' section. A red arrow points to the 'X\_Axis' tag, which is linked to the 'ud\_CMAX\_Control' data type.

**Data Type Details (Center Panel):** The 'ud\_CMAX\_Control' data type is defined with the following members:

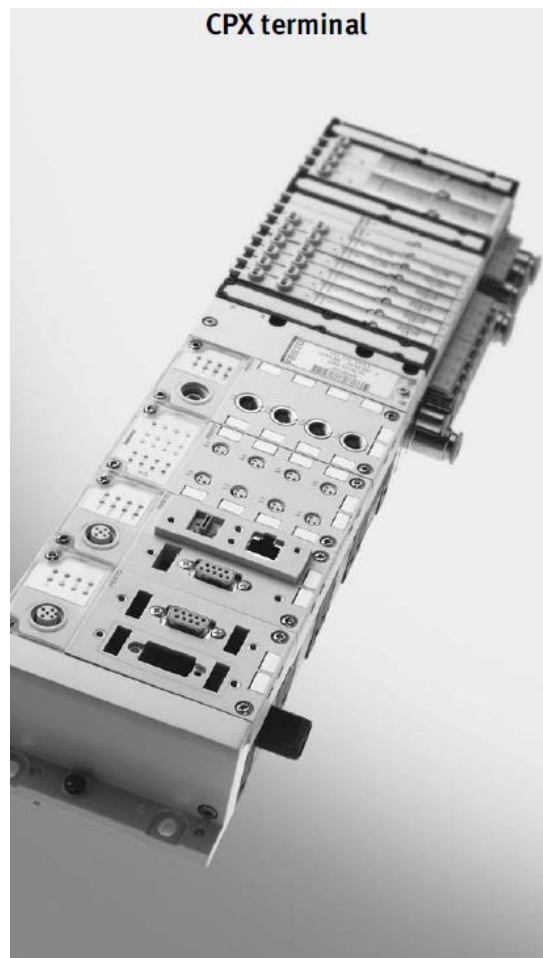
Name	Data Type	Style	Description	External Access
FB	CMAX_FB_CTRL		Function Block AOI	Read/Write
Mode	ud_CMAX_Modes			Read/Write
EnableDrive	BOOL	Decimal	Create Drive Ready	Read/Write
Fault_Reset	BOOL	Decimal	Reset Existing Faults	Read/Write
ONS	DINT	Decimal	One Shot	Read/Write
HomeSeqStart	BOOL	Decimal	Debug Bit Home Sequence Start	Read/Write
HomeSeqDelay	TIMER		Home Sequence Reset Delay	Read/Write
HomeSeqStarted	BOOL	Decimal	Home Sequence Started	Read/Write
HomeSeqMC	BOOL	Decimal	Home Sequence Complete	Read/Write
Current_Position	REAL	Float	Current Position (Scaled)	Read/Write
Motion	CMAX_MAM[9]			Read/Write
Safety	BOOL	Decimal		Read/Write
RecovReq	BOOL	Decimal		Read/Write
PNU_224_1_R	CMAX_FB_FFC			Read/Write
PNU_1192_7_w	CMAX_FB_FFC			Read/Write
PNU_1192_7_R	CMAX_FB_FFC			Read/Write
PNU_1174_1_R	CMAX_FB_FFC			Read/Write
PNU_1170_1_R	CMAX_FB_FFC			Read/Write
MoveTest	ud_Stepper			Read/Write
IdentStep	ud_Stepper			Read/Write
MoveStep	ud_Stepper			Read/Write
TeachStep	ud_Stepper			Read/Write

**Controller Tags (Bottom Panel):** The 'X\_Axis' tag is defined with the following members:

Name	Description	Data Type
X_Axis	X Axis	ud_CMAX_Control
X_Axis.FB	X Axis Function Block AOI	CMAX_FB_CTRL
X_Axis.Mode	X Axis	ud_CMAX_Modes
X_Axis.EnableDrive	X Axis Create Drive Ready	BOOL
X_Axis.Fault_Reset	X Axis Reset Existing Faults	BOOL
X_Axis.ONS	X Axis One Shot	DINT
X_Axis.HomeSeqStart	X Axis Debug Bit Home Sequence Start	BOOL
X_Axis.HomeSeqDelay	X Axis Home Sequence Reset Delay	TIMER
X_Axis.HomeSeqStarted	X Axis Home Sequence Started	BOOL
X_Axis.HomeSeqMC	X Axis Home Sequence Complete	BOOL
X_Axis.Current_Position	X Axis Current Position (Scaled)	REAL
X_Axis.Motion	X Axis CMAX Motion Axis Move (Position)	CMAX_MAM[9]
X_Axis.Safety	X Axis Pre-Move Safety	BOOL
X_Axis.RecovReq	X Axis Error Recovery Required	BOOL
X_Axis.PNU_224_1_R	X Axis PNU_224_1 READ Get Error Code On DISPLAY of CMAX	CMAX_FB_FFC
X_Axis.PNU_1192_7_w	X Axis PNU_1192_7 WRITE Movement Test Status	CMAX_FB_FFC
X_Axis.PNU_1192_7_R	X Axis PNU_1192_7 READ Movement Test Status	CMAX_FB_FFC
X_Axis.PNU_1174_1_R	X Axis PNU_1174_1 READ ONLY Movement Test Status	CMAX_FB_FFC
X_Axis.PNU_1170_1_R	X Axis PNU_1170_1 READ Get Identification Type=0 Dynamic Type=1 Static	CMAX_FB_FFC
X_Axis.MoveTest	X Axis Movement Test Seq	ud_Stepper
X_Axis.IdentStep	X Axis Identification Seq	ud_Stepper
X_Axis.MoveStep	X Axis Move Cycle Seq	ud_Stepper
X_Axis.TeachStep	X Axis Teach Record Seq	ud_Stepper

## FHPP for the CMAX (Festo Handling and Position Profile)

The following reference information was taken from the Festo “CPX-CMAX FHPP for Control 559757g1 0908NH.pdf”



**FESTO**

**Description**  
Communication  
profile

FHPP for the  
CMAX axis  
controller

Activation and  
diagnostics via  
CPX node

Typ CPX-CMAX-C1-1



**Description**  
559 757  
en 0908NH  
[727 411]

## FHPP for the CMAX (Festo Handling and Position Profile) Overview

FHPP is used for controlling, diagnosing, commissioning and parameterizing the CPX-CMAX servo pneumatic controller via the Fieldbus.

Term/abbreviation	Meaning
Festo Handling and Positioning Profile (FHPP)	Uniform fieldbus data profile for position controllers
Festo Parameter Channel FPC (Festo Parameter Channel)	Specific Parameter Access as per the "Festo Handling and Positioning Profile"

Festo has developed an optimised data profile especially tailored to the target applications for handling and positioning tasks called the "Festo Handling and Positioning Profile (FHPP)". The FHPP enables uniform control and programming for the various fieldbus systems and controllers from Festo.

There are 4 Operating Modes:

- ☐ Record Select
- ☐ Direct
- ☐ Commissioning
- ☐ Parameterizing

Data	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
<b>Output data</b>	Bytes 1 and 2 (fixed) are retained in every operating mode (except byte 2 for parametrisation). They contain control and status bytes (e.g. CCON, SCON, ...) for enabling the CMAX and setting the operating modes		Bytes 3 to 8 depend on the selected operating mode (direct mode, record select) and transmit further control and status bytes (e.g. CDIR, SDIR,...), as well as setpoint and actual values: <ul style="list-style-type: none"> <li>Record number or setpoint position in the output data</li> <li>Feedback of actual position and record number in the input data</li> <li>Additional operating mode- and control mode-dependent setpoint and actual values</li> </ul>					
<b>Input data</b>								



## Festo CPX Terminal – Rules of Addressing

- ❑ The address assignment of the inputs does not depend on the address assignment of the outputs.
- ❑ The I/Os of various module types are assigned separately from each other. The following order applies:

Sequence of addressing		Description
1.	I/O diagnostic interface <sup>1)</sup>	Can be activated by DIL switch. When activated, it will occupy the first 16 inputs and outputs in the address range.
2.	Analogue modules	Modules with analogue inputs/outputs
3.	Technology module	e.g. CP interface, front-end controller CPX-FEC
4.	Digital modules	Modules with digital inputs/outputs
<sup>1)</sup> In exceptional cases this address range can also be occupied by status bits (see Tab. 1/3).		

# CMAX\_FB\_CTRL Master Control Function Block (AOI)

This Function Block (AOI) is used to control all possible functions of the CPX-CMAX controller. This configured AOI instruction can operate the controller as a standalone for testing purposes and controller understanding or with all the additionally provided code.

With this module, it is possible to control the axis controller of type CPX-CMAX-C1-1 in the following six combinations of the operating type and control type:

- ☐ Record select mode
- ☐ Direct mode, position control – Free profile
- ☐ Direct mode, position control – Auto profile
- ☐ Direct mode, pressure/force control
- ☐ Parameterization
- ☐ Commissioning

**NOTE:** The AOI(Add On Instruction) itself needs to be on a rung that has power at all times or it will not function correct. The only instruction that should be used prior to the AOI is a bit that monitors the active communications between the PLC and the CPX Fieldbus module where the CPX-CPX-CMAX(Servo Pneumatic Controller) is installed.



The screenshot shows the 'Controller Organizer' window with the 'Data Types' section expanded. The 'ud\_CMAX\_Control' data type is selected. A red arrow points from this selection to the 'Name' field in the configuration dialog. The dialog also shows a 'Description' field and a 'Members' list. The 'Members' list contains the following data:

Name	Data Type	Style	Description	External Access
FB	CMAX_FB_CTRL		Function Block AOI	Read/Write
Mode	ud_CMAX_Mode			Read/Write
EnableDrive	BOOL	Decimal	Enable Drive Ready	Read/Write
Fault_Reset	BOOL	Decimal	Reset Existing Faults	Read/Write
ONS	DINT	Decimal	One Shot	Read/Write
HomeSeqStart	BOOL	Decimal	Debug Bit Home Sequence Start	Read/Write
HomeSeqDelay	TIMER		Home Sequence Reset Delay	Read/Write
HomeSeqStarted	BOOL	Decimal	Home Sequence Started	Read/Write
HomeSeqMC	BOOL	Decimal	Home Sequence Complete	Read/Write
Current_Position	REAL	Float	Current Position (Scaled)	Read/Write
Motion	CMAX_MAM[9]			Read/Write
Safety	BOOL	Decimal	Pre-Move Safety	Read/Write
RecoveryReq	BOOL	Decimal	Error Recovery Required	Read/Write
PNU_224_1_R	CMAX_FB_FFC		PNU 224.1 READ Get Error Code On DISPLAY of CMAX	Read/Write
PNU_1192_7_W	CMAX_FB_FFC		PNU 1192.7 WRITE Movement Test Status	Read/Write
PNU_1192_7_R	CMAX_FB_FFC		PNU 1192.7 READ	Read/Write
PNU_1174_1_R	CMAX_FB_FFC		PNU 1174.1 READ	Read/Write
PNU_1170_1_R	CMAX_FB_FFC		PNU 1170.1 READ	Read/Write
MoveTest	ud_Stepper		Movement Test	Read/Write
IdentStep	ud_Stepper		Identification Step	Read/Write
MoveStep	ud_Stepper		Move Cycle Seq	Read/Write
TeachStep	ud_Stepper		Teach Record S	Read/Write

CMAX_FB_CTRL	
CMAX Master Control Function Block (AOI)	
<b>CMAX_FB_CTRL</b>	<b>X_Axis FB 0</b>
I_Data	FB32_CMAXIData
I_Data_ArrayStartAddr	0
Q_Data	FB32_CMAXQData
Q_Data_ArrayStartAddr	0
Control_O_Arte_a	0
CCON_Enable	1
CCON_Stop	1
CCON_BrakeRelease	1
CCON_ResetFault	0
CCON_LockFCTAccess	0
CCON_OPM1_OperationMode	1
CCON_OPM2_OperationMode	0
CPDS_StartTask	0
CPDS_StartHoming	0
CPDS_LogPositive	0
CPDS_LogNegative	0

Scope: AB_FB32_CMAx		Show: All Tags
Name	Description	Data Type
X_Axis	X Axis	ud_CMAX_Control
+ X_Axis.FB	X Axis Function Block ADI	CMAx_FB_CTRL
+ X_Axis.Mode	X Axis	ud_CMAX_Modes
- X_Axis.EnableDrive	X Axis Create Drive Ready	BOOL
- X_Axis.Fault_Reset	X Axis Reset Existing Faults	BOOL
+ X_Axis.ONS	X Axis One Shot	DINT
- X_Axis.HomeSeqStart	X Axis Debug Bit Home Sequence Start	BOOL
+ X_Axis.HomeSeqDelay	X Axis Home Sequence Reset Delay	TIMER
- X_Axis.HomeSeqStarted	X Axis Home Sequence Started	BOOL
- X_Axis.HomeSeqMC	X Axis Home Sequence Complete	BOOL
- X_Axis.Current_Position	X Axis Current Position (Scaled)	REAL
+ X_Axis.Motion	X Axis CMAX Motion Axis Move (Position)	CMAx_MAM[3]
- X_Axis.Safety	X Axis Pre-Move Safety	BOOL
- X_Axis.RecovReq	X Axis Error Recovery Required	BOOL
+ X_Axis.PNU_224_1_R	X Axis PNU_224_1 READ Get Error Code On DISPLAY of CMAx	CMAx_FB_FPC
+ X_Axis.PNU_1192_7_W	X Axis PNU_1192_7 WRITE Movement Test Status	CMAx_FB_FPC
+ X_Axis.PNU_1192_7_R	X Axis PNU_1192_7 READ Movement Test Status	CMAx_FB_FPC
+ X_Axis.PNU_1174_1_R	X Axis PNU_1174_1 READ ONLY Movement Test Status	CMAx_FB_FPC
+ X_Axis.PNU_1170_1_R	X Axis PNU_1170_1 READ Get Identification Type=0 Dynamic Type=1 Static	CMAx_FB_FPC
+ X_Axis.MoveTest	X Axis Movement Test Seq	ud_Stepper
+ X_Axis.IdentStep	X Axis Identification Seq	ud_Stepper
+ X_Axis.MoveStep	X Axis Move Cycle Seq	ud_Stepper
+ X_Axis.TeachStep	X Axis Teach Record Seq	ud_Stepper

# CMAX\_FB\_CTRL I/O Data Assignment

When the user has configured the Generic Ethernet Module and the data size for the entire CPX terminal, the data for the controller (8 Bytes Input/Output) will reside “somewhere” among the other CPX terminal data.

The NAME provided for this Module is the name of the tag found in the controller tags used for IO (Inputs/Outputs) to the CPX Terminal/Rack. Enter these tag names in the AOI as shown for the I\_Data and O\_Data Parameters

Due to the addressing rules of the CPX terminal, the data for the CPX-CMAX can reside anywhere amongst other installed modules (See Festo CPX – Rules of Addressing page later in this document).

For this reason, this AOI has user inputs for the “Array Number” where the mapping of Inputs and Outputs Starts/Begins (I\_Data\_ArrayStartAddr, O\_Data\_ArrayStartAddr). For this sample, the starting array address is “0” as shown - Enter your starting array number for your application.

At this point, the user could go online and begin to operate the controller by entering data in the instruction itself if they desired.

**Controller Organizer**

- Add-On Instructions
  - CMAX\_FB\_CTRL
  - CMAX\_FB\_FPC
  - CMAX\_MAM
- Data Types
  - User-Defined
    - ac\_Flasher
    - a\_Date
    - a\_Time
    - ud\_CMAX\_Control
    - ud\_CMAX\_Modes
    - ud\_PlcStatusBits
    - ud\_Stepper
    - z\_Version
- I/O Configuration
  - 1756 Backplane, 1756-A7
    - [0] 1756-L61 AB\_FB32\_CMAX\_V19\_AOI\_Cust
    - [1] 1756-ENBT/A ENET\_IP
      - Ethernet
        - 1756-ENBT/A ENET\_IP
        - ETHERNET-MODULE FB32\_CMAX

**Scope: AB\_FB32\_CMAX Show: All Tags**

Name	Description	Data Type
FB32_CMAX:O		AB:ETHERNET_MODULE_SINT_8Bytes:0:0
FB32_CMAX:O.Data		SINT[8]
FB32_CMAX:O.Data[0]	#1: CMAX-C1-1 - [08] CMAX Control Byte 1	SINT
FB32_CMAX:O.Data[1]	#1: CMAX-C1-1 - [09] CMAX Control Byte 2	SINT
FB32_CMAX:O.Data[2]	#1: CMAX-C1-1 - [10] CMAX Control Byte 3	SINT
FB32_CMAX:O.Data[3]	#1: CMAX-C1-1 - [11] CMAX Control Byte 4	SINT
FB32_CMAX:O.Data[4]	#1: CMAX-C1-1 - [12] CMAX Control Byte 5	SINT
FB32_CMAX:O.Data[5]	#1: CMAX-C1-1 - [13] CMAX Control Byte 6	SINT
FB32_CMAX:O.Data[6]	#1: CMAX-C1-1 - [14] CMAX Control Byte 7	SINT
FB32_CMAX:O.Data[7]	#1: CMAX-C1-1 - [15] CMAX Control Byte 8	SINT
FB32_CMAX:I		AB:ETHERNET_MODULE_SINT_8Bytes:1:0
FB32_CMAX:I.Data		SINT[8]
FB32_CMAX:I.Data[0]	#1: CMAX-C1-1 - [10] CMAX Status Byte 1	SINT
FB32_CMAX:I.Data[1]	#1: CMAX-C1-1 - [11] CMAX Status Byte 2	SINT
FB32_CMAX:I.Data[2]	#1: CMAX-C1-1 - [12] CMAX Status Byte 3	SINT
FB32_CMAX:I.Data[3]	#1: CMAX-C1-1 - [13] CMAX Status Byte 4	SINT
FB32_CMAX:I.Data[4]	#1: CMAX-C1-1 - [14] CMAX Status Byte 5	SINT
FB32_CMAX:I.Data[5]	#1: CMAX-C1-1 - [15] CMAX Status Byte 6	SINT
FB32_CMAX:I.Data[6]	#1: CMAX-C1-1 - [16] CMAX Status Byte 7	SINT
FB32_CMAX:I.Data[7]	#1: CMAX-C1-1 - [17] CMAX Status Byte 8	SINT
FB32_CMAX:C		AB:ETHERNET_MODULE:C:0

**CMAX\_FB\_CTRL**

- CMAX: Master Control Function Block (AOI)
  - CMAX\_FB\_CTRL
    - I\_Data\_ArrayStartAddr FB32\_CMAX:I.Data
    - O\_Data FB32\_CMAX:O.Data
    - O\_Data\_ArrayStartAddr
    - C.ControlActive
    - CCOM\_Enable
    - CCOM\_Stop
    - CCOM\_BrakeRelease
    - CCOM\_ResetFault
    - CCOM\_LookForAccess
    - CCOM\_OPM1\_OperationMode
    - CCOM\_OPM2\_OperationMode
    - CPOS\_StartTask
    - CPOS\_StartHoming
    - CPOS\_JogPositive
    - CPOS\_JogNegative
    - CPOS\_TeachActualValue
    - CDIR\_RelativeMoveReq
    - CDIR\_PositionCtrl0\_ForceCtrl
    - CDIR\_PositionCtrlFree0\_AutoI
    - CDIR\_TrackingEnable
    - CDIR\_VelocityLimitDisable
    - CDIR\_StrokeLimitDisable
    - CDIR\_StopExec0\_FastI
    - RecordNumReq
    - RecordNumStatus
    - RecordMode\_PrimaryValue
    - DirectMode\_SecondarySetPoint
    - DirectMode\_SecondaryValue
    - RecordMode\_PrimaryValue
    - ComMode\_Parameter1
    - ComMode\_Parameter2
    - ComMode\_Function

## CMAX\_FB\_CTRL Parameters Description Legend

Legend	
<b>Input / Output</b>	Parameter Name of an input/output of the AOI (Add-On Instruction)
<b>Type</b>	Data Type used for the Parameter.
<b>Description</b>	Brief description of the input/output options at an input/output of the AOI (Add-On Instruction) (0 = FALSE, 1 = TRUE)
<b>Use</b>	Operating mode where this input/output of the AOI (Add-On Instruction) is used.

## CMAX\_FB\_CTRL Description of Parameters I/O Data

Input / output	Type	Description	Use
I_Data	SINT[8]	Array Input Tag Name	All operating modes
I_Data_ArrayStartAddr	SINT	Array Number Where the Mapping of Inputs begin	All operating modes
O_Data	SINT[8]	Array Output Tag Name	All operating modes
O_Data_ArrayStartAddr	SINT	Array Number Where the Mapping of Outputs begin	All operating modes

# CMAX\_FB\_CTRL Description of Parameters Control Area CCON

C o n t r o l A r e a					
FHPP Manual		Input / output	Description		
BYTE 1 CCON	B0 ENABLE	CCON_Enable	= 0: Drive (controller) blocked = 1: Enable drive (controller)		
	B1 STOP	CCON_Stop	= 0: Stop active (execute stop ramp + cancel positioning job). The drive stops with a stop ramp, the job is terminated, standstill monitoring is deactivated. = 1: Enable operation. Not permitted in parametrisation mode. If there is a logic 1 in parametrisation mode, a warning is signaled.		
	B2 BRAKE	CCON_BrakeRelease	= 0: Activate brake (0 V at brake output) = 1: Release brake (24 V at brake output) Note: The allocation can be inverted by means of parametrisation (PNU 522:02). If the enable and brake are activated, CMAX executes force control with force setpoint 0.		
	B3 RESET	CCON_ResetFault	With a <b>rising edge</b> , a registered fault signal is cleared and the fault state is exited if successful.		
	B5 LOCK	CCON_LockFCTAccess	Access to the service interface (via FCT): = 1: FCT may only observe. Not possible to take over the device control (FCT). = 0: FCT can take over the device control (in order to change parameters or to control inputs).		
	B6 OPM1	CCON_OPM1_OperationMode	Bit	OPM2	Operating mode
				0	Record select mode
				0	Direct mode
				1	Commissioning mode
	B7 OPM2	CCON_OPM2_OperationMode		1	Parametrising mode

## CMAX\_FB\_CTRL Description of Parameters **Control Area CPOS**

<b>C _ o n t r o l _ A r e a</b>			
FHPP Manual		Input / output	Description
<b>BYTE 2 CPOS</b>	<b>B1 START</b>	<b>CPOS_StartTask</b>	With a <b>rising edge</b> the current nominal values will be transferred and positioning started.
	<b>B2 HOM</b>	<b>CPOS_StartHoming</b>	With a <b>rising edge</b> , homing is started with the set parameters and referencing is reset. A malfunction is signaled with an absolute measuring system.
	<b>B3 JOGP</b>	<b>CPOS_JogPositive</b>	The drive moves at the specified speed in the direction of larger actual values, providing the bit is set.
	<b>B4 JOGN</b>	<b>CPOS_JogNegative</b>	The drive moves at the specified speed toward smaller actual values as long as the bit is set. If JoggingPos and JoggingNeg are set at the same time, the drive moves in the positive direction.
	<b>B5 TEACH</b>	<b>CPOS_TeachActualValue</b>	At a <b>falling edge</b> , the current actual value is imported into the setpoint register of the currently addressed positioning record.



# CMAX\_FB\_CTRL Description of Parameters Control Area CDIR

C o n t r o l A r e a			
FHPP Manual		Input / output	Description
BYTE 3 CDIR	B0 ABS	CDIR_RelativeMoveReq	= 0: Setpoint value is absolute (with respect to the project zero point) = 1: Setpoint value is relative to the last setpoint/actual value (for MC) or to the actual value (if there is no MC). Force jobs which follow positioning jobs refer to the force 0.
	B1 COM1	CDIR_ForceCtrlEnable	= 0: Position control = 1: Force control
	B2 COM2	CDIR_VelocityCtrlAutoEnable	For position control only: = 0: Free profile: Speed and acceleration are freely specified = 1: Auto profile: Speed and accelerations are specified by the controller (Speed and accelerations are selected by the controller according to the identification so that the target position is reached as quickly as possible without overshooting.) For logic 1 with force control mode, a malfunction is signaled.
	B3 CONT	CDIR_TrackingEnable	For position control: Activates continuous tracking mode (continuous setpoint specification): = 0: Do not activate continuous tracking mode = 1: Activate continuous tracking mode
	B4 YLIM	CDIR_VelocityLimitDisable	For force control: = 0: Activate speed limit = 1: Deactivate speed limit
	B5 XLIM	CDIR_StrokeLimitDisable	For force control: = 0: Activate stroke monitoring = 1: Deactivate stroke monitoring
	B6 FAST	CDIR_FastStopEnable	Control when the target setpoint value has been reached: (MC is only set when the job is completed according to the quality class. In the case of a fast stop, standstill monitoring is deactivated.) = 0: Exact stop = 1: Fast stop

# CMAX\_FB\_CTRL Description of Parameters Control Area SCON

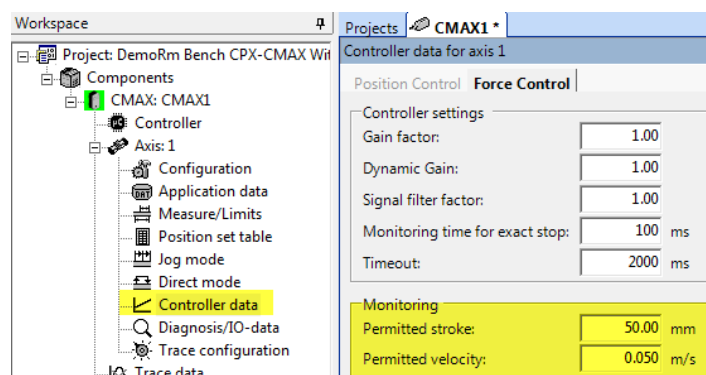
C o n t r o l A r e a						
FHPP Manual		Input / output	Description			
BYTE 1 SCON	B0 ENABLED	SCON_Enabled	= 0: Drive/controller disabled, controller not active = 1: Drive / controller enabled			
	B1 OPEN	SCON_OperationEnabled	= 0: Stop active = 1: Operation enabled, positioning possible			
	B2 WARN	SCON_WarningPresent	= 0: Warning not registered = 1: Warning registered			
	B3 FAULT	SCON_FaultPresent	= 0: No malfunction = 1: There is a malfunction or a malfunction reaction is active.			
	B4 24VL	SCON_LoadVoltageApplied	= 0: No load voltage = 1: Load voltage applied			
	B5 FCT_MMI	SCON_FCTDeviceControlled	= 0: Device control free (e.g. PLC/fieldbus) = 1: Device control by software (FCT)			
	B6 OPM1	SCON_OPM1_OperationState	Bit	OPM2	OPM1	Operating mode
				0	0	Record select mode
	B7 OPM2	SCON_OPM2_OperationState		0	1	Direct mode
				1	0	Commissioning mode
			1	1	Parametrising mode	

# CMAX\_FB\_CTRL Description of Parameters **Control Area SPOS**

<b>C o n t r o l A r e a</b>			
FHPP Manual		Input / output	Description
<b>BYTE 2 SPOS</b>	<b>B1 ACK</b>	<b>SPOS_AcknowledgeStart</b>	= 0: Ready for start = 1: Start executed (When programming the handshakes of StartTask and AckStart, any registered malfunctions must always be taken into account since AckStart is not set if there is a malfunction.)
	<b>B2 MC</b>	<b>SPOS_MotionComplete</b>	= 0: Positioning job active = 1: Positioning job completed, where applicable with error (MC is set for the first time after switch-on [status "Drive disabled"])
	<b>B3 TEACH</b>	<b>SPOS_AcknowledgeTeach</b>	= 0: Teaching carried out, actual value is transferred = 1: Ready for teaching
	<b>B4 MOV</b>	<b>SPOS_AxisIsMoving</b>	Movement monitoring. = 0: Drive is not moving (axis speed signal < limit) = 1: Drive is moving
	<b>B5 DEV</b>	<b>SPOS_FollowingError</b>	Following error or tolerance monitoring. = 0: No following error / within tolerance = 1: Following error active / outside of tolerance
	<b>B6 STILL</b>	<b>SPOS_StandstillWarning</b>	Standstill monitoring. = 0: Standstill warning not active = 1: Standstill warning active, drive moved after MC (position control)
	<b>B7 REF</b>	<b>SPOS_DriveReferenced</b>	= 0: Referencing must be carried out = 1: Reference information present, homing not necessary

# CMAX\_FB\_CTRL Description of Parameters **Control Area SDIR**

<b>C o n t r o l A r e a</b>			
FHPP Manual		Input / output	Description
<b>BYTE 3 SDIR</b>	<b>B0 ABS</b>	<b>SDIR_RelativeMoveState</b>	= 0: Setpoint value is absolute (Relevant only after move Start) = 1: Setpoint value is relative to last setpoint value (Relevant only after move Start)
	<b>B1 COM1</b>	<b>SDIR_ForceCtrl_State</b>	= 0: Position control active = 1: Force control active
	<b>B2 COM2</b>	<b>SDIR_VelocityCtrlAuto_State</b>	For position control only: = 0: Free profile = 1: Auto profile
	<b>B3 CONT</b>	<b>SDIR_TrackingStatus</b>	Continuous tracking mode acknowledgment (continuous setpoint specification): = 0: Continuous tracking mode not active = 1: Continuous tracking mode active
	<b>B4 VLIM</b>	<b>SDIR_VelocityLimitReached</b>	For force control only: = 0: Speed limit not reached = 1: Speed limit reached
	<b>B5 XLIM</b>	<b>SDIR_StrokeLimitReached</b>	For force control only: = 0: Stroke limit not reached = 1: Stroke limit reached, malfunction is signaled
	<b>B6 FAST</b>	<b>SDIR_FastStop_State</b>	= 0: Exact stop is active (Relevant only after move Start) = 1: Fast stop is active, malfunction is signaled (Relevant only after move Start)



# CMAX\_FB\_CTRL Description of Parameters **Control Area RSB**

<b>C_o_n_t_r_o_l_A_r_e_a</b>			
FHPP Manual		Input / output	Description
<b>BYTE 4 RSB</b>	<b>B0 RC1</b>	<b>RSB_RecordChainActive</b>	If at least one step criterion has been configured: = 0: The first step criterion has not yet been reached. = 1: The first switch was executed.
	<b>B1 RCC</b>	<b>RSB_RecordChainComplete</b>	If at least one step criterion has been configured and Motion Complete (MC) applies: = 0: Switching condition not met, record chaining aborted. = 1: Record chain was processed to the end of the chain.
	<b>B2 COM1</b>	<b>RSB_Position_Force_Status</b>	Position Control Active=0 Force Control Active=1
	<b>B3 RCE</b>	<b>RSB_RecordChainError</b>	If at least one step criterion has been configured: = 0: No error during record switching or no record switching programmed = 1: Record switching was programmed, but not executed. Record chaining aborted. A malfunction is signaled. (Record switching: Position set sequencing is equivalent to record chaining)
	<b>B4 VLIM</b>	<b>RSB_VelocityLimitReached</b>	Force Control Velocity Limit Reached
	<b>B5 XLIM</b>	<b>RSB_StrokeLimitReached</b>	Force Control Stroke Limit Reached

## CMAX\_FB\_CTRL Description of Parameters **Record Mode**

<b>R_e_c_o_r_d_M_o_d_e_A_c_t_i_v_e</b>			
FHPP Manual		Input / output	Description
<b>BYTE 3</b>	<b>Record no.</b>	<b>RecordNumReq</b>	Record number of the record to be started (1 ... 64)
<b>BYTE 3</b>	<b>Record no.</b>	<b>RecordNumStatus</b>	Feedback of last started record (1 to 64) (In a record chain, the actual record number always contains the number of the record which is currently being executed. It changes during record switching, so without a starting edge.)
<b>BYTE 5-8</b>	<b>Primary actual value</b>	<b>RecordMode_PrimaryValue</b>	Primary actual value (Actual value of the position or force in the set system of units depending on the operating status and the FHPP setting (PNU 523).)

# CMAX\_FB\_CTRL Description of Parameters **Direct Mode**

<b>D_i_r_e_c_t_M_o_d_e_A_c_t_i_v_e</b>			
FHPP Manual		Input / output	Description
<b>BYTE 4</b>	<b>Secondary Setpoint</b>	<b>DirectMode_SecondarySetPoint</b>	<p>Secondary setpoint                      ('Setpoint value, depending on control mode and FHPP setting (PNU 523)                      - Speed in percent of the default value (PNU 540),                      - Force ramp in percent of the default value (PNU 550),                      - Workpiece mass in percent of the default value (PNU 544 or PNU 551).                      Value range 0 to 100, no sign. Impermissible setpoint values are limited.                      For secondary setpoint "Workpiece mass", 100% of the basic value is always used for the speed or force ramp.                      The setpoint is accepted when there is a positive edge at StartTask.)</p>
<b>BYTE 5-8</b>	<b>Primary Setpoint</b>	<b>DirectMode_PrimarySetpoint</b>	<p>Primary setpoint                      (Setpoint value of the position or force in the set system of units. The setpoint value is accepted with a positive edge at StartTask.                      In the case of continuous tracking mode, the setpoint position is accepted continuously after the start until the end of continuous tracking mode.)</p>
<b>BYTE 4</b>	<b>Secondary Actual Value</b>	<b>DirectMode_SecondaryValue</b>	<p>Secondary actual value                      (Actual value of the speed in percent of the default value (PNU540).                      The secondary actual value for the speed has a sign, so positive and negative values can be displayed. The entire value range is utilised, i.e. the displayed speed lies in the range from -128 % to +127 %. Higher speeds are limited to -128 % or +127 %.)</p>
<b>BYTE 5-8</b>	<b>Primary Actual Value</b>	<b>DirectMode_PrimaryValue</b>	<p>Primary actual value                      (Actual value of the position or force in the set system of units depending on the operating status and the FHPP setting (PNU 523).)</p>

# CMAX\_FB\_CTRL Description of Parameters **Commission Mode**

<b>C o m m i s s i o n M o d e A c t i v e</b>			
FHPP Manual		Input / output	Description
BYTE 4	Param. 1	ComMode_Parameter1	Parameter 1 (When executing a commissioning operation: Reserved = 0. During teaching: Teach target. A zero (=0) must be transferred to unused setpoint bytes.)
BYTE 5-8	Param. 2	ComMode_Parameter2	Parameter 2 (Only for commissioning operation "Identification": Current workpiece mass in the set system of units. A zero (=0) must be transferred to unused setpoint bytes.)
BYTE 3	Function	ComMode_Function	Function <i>[Error state Default]</i>
BYTE 3	Function	ComMode_FunctionStatus	Feedback of the commissioning operation currently being executed.
BYTE 4	Progress	ComMode_Progress	Progress (When executing a commissioning operation: The progress display in the status data indicates how far along the function has already progressed for long operations. Display in percent (0% to 100%). Jumps can occur in the progress display (e.g. from 24% to 60%). At the end of the function, the progress counter is set to 255 (0xFF). During teaching: Teach target.)
BYTE 5-8	Primary actual value	ComMode_PrimaryValue	Primary actual value (Actual value of the position or force in the set system of units depending on the operating status and the FHPP setting (PNU 523).)

Note	Commissioning Operation to be started with rising edge of StartTask in Commissioning mode			
	Value	Description	Param.1	Param.2
	0	Impermissible	-	-
	1	Execute identification run	0	Workpiece
	2	Execute movement test	0	0
	3 to 255	Impermissible	-	-



# CMAX FB\_CTRL Description of Parameters **Parameter Mode**

<b>P_a_r_a_m_e_t_e_r_M_o_d_e_A_c_t_i_v_e</b>			
FHPP Manual	Input / output	Description	
	<b>Req_Enable</b>	The transmission of a parameter is started by a <b>rising edge</b>	
	<b>Req_Complete</b>	= 0: Transmission was not initiated = 1: Transmission was Completed	
	<b>Req_Failed</b>	= 0: No error while transmitting a parameter = 1: Error transmitting a parameter	
<b>BYTE 3-4</b>	<b>Param. Identifier ReqID (Bits12-15)</b>	<b>Req_ID</b>	= 6: Read parameters = 8: Write parameters
	<b>Param. Identifier PNU (Bits0-11)</b>	<b>Req_PNU</b>	Parameter number of the corresponding parameter
<b>BYTE 3</b>	<b>Subindex</b>	<b>Req_SubIndex</b>	Subindex of the corresponding parameter
<b>BYTE 5-8</b>	<b>Param. Value</b>	<b>Req_Value</b>	Parameter value when 'WRITING' a parameter
<b>BYTE 3-4</b>	<b>Param. Identifier ResID (Bits12-15)</b>	<b>Res_ID</b>	= 6: Read parameters = 8: Write parameters
	<b>Param. Identifier PNU (Bits0-11)</b>	<b>Res_PNU</b>	Display of the current parameter number
<b>BYTE 3</b>	<b>Subindex</b>	<b>Res_SubIndex</b>	Display of the current subindex
<b>BYTE 5-8</b>	<b>Param. Value</b>	<b>Res_Value</b>	Response Value = 0: No Reply = 5: Parameter Value was transferred = 7: Error/Parameter request Failed
		<b>Res_FaultCode (See Note 2)</b>	Parameter Transmission Errors

Parameter Transmission Errors (Res_FaultCode)	
Error	Error description
0	Impermissible PNU.
1	Parameter value cannot be changed.
2	Lower or upper value limit exceeded.
3	Invalid sub index.
11	No higher-order controller. FCT must accept device control in order to write this parameter. This error can only be generated via the service interface.
12	The password entered is wrong.
17	Request cannot be carried out due to operating status. Please check operating mode, stop and enable signals.
101	Request ID is not supported.
102	Parameter cannot be read (password).
103	The system of measurement has not been configured yet. Access to the parameter is not possible.
104	The cylinder type has not been configured yet. Access to the parameter is not possible.
105	The system of measurement has already been configured and cannot be changed without a data reset.
106	Cylinder type cannot be changed since it does not match the system of measurement.
107	The value cannot be changed since there is identification data. Please reset the identification data before writing the value.
108	The parameter value does not match the hardware recognised. (Note: the cylinder type must match the measuring system)
109	Serial numbers cannot be changed before the identification data has been reset.

## CMAX\_FB\_CTRL Establish Ready Status

Assuming that the CPX terminal has been switched on, the load voltage is present and the PLC is in control, the following actions are to be executed to establish the complete ready status:

User action	Acknowledgement
No action	SCON_LoadVoltageApplied = 1 SPOS_MotionComplete = 1
CCON_Enable = 1	SCON_Enabled = 1
CCON_Stop = 1	SCON_OperationEnabled = 1

## Homing

User action	Acknowledgement
CPOS_StartHoming = 0 → 1	SPOS_AcknowledgeStart = 1, SPOS_MotionComplete = 0, SPOS_AxisIsMoving = 1 (depending on the starting position)
CPOS_StartHoming = 0	After completing homing: SPOS_AxisIsMoving = 0, SPOS_MotionComplete = 1, SPOS_DriveReferenced = 1, SPOS_AcknowledgeStart = 0

## CMAX\_FB\_CTRL Record Select Mode

A position set table can be configured with FCT (Festo Configuration Tool Software) to store up to 64 records which contain all the necessary information for a positioning task within the CPX-CMAX controller itself. When the user has placed the CMAX controller in this mode, the PLC selects & then executes the record to be used from the stored list.

Data	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Output data	CCON	CPOS	Record no.	Reserved	Reserved			
Input data	SCON	SPOS	Record no.	RSB	Primary actual value (actual position, actual force)			

Assignment of the **control bytes (record select mode)**

CCON Byte 1	B7 OPM2	B6 OPM1	B5 LOCK	B4 –	B3 RESET	B2 BRAKE	B1 STOP	B0 ENABLE
	Mode selection		Block software access	–	Reset fault	Release brake	Stop	Enable drive
CPOS Byte 2	B7 –	B6 CLEAR	B5 TEACH	B4 JOGN	B3 JOGP	B2 HOM	B1 START	B0 HALT
	–	–	Teach value	Jog negative	Jog positive	Start homing	Start positioning task	–
Record no. Byte 3	Byte 3: Record number of the starting record (1 to 64)							
res. Bytes 4 to 8	Reserved = 0							

Assignment of the **status bytes (record select mode)**

SCON Byte 1	B7 OPM2	B6 OPM1	B5 FCT_MMI	B4 24VL	B3 FAULT	B2 WARN	B1 OPEN	B0 ENABLED
	Operating mode acknowledgement		Software device control	Load voltage applied	Fault	Warning	Operation enabled	Drive enabled
SPOS Byte 2	B7 REF	B6 STILL	B5 DEV	B4 MOV	B3 TEACH	B2 MC	B1 ACK	B0 HALT
	Drive referenced	Standstill warning	Following error	Axis is moving	Acknowledge Teach	Motion Complete	Acknowledge Start	–
Record no. Byte 3	Acknowledgement of the last started record (1 to 64) In the case of record chaining, the actual record number always contains the currently actually executed record number. This changes, then, when the record is switched without a starting edge.							
RSB Byte 4	B7 –	B6 –	B5 XLIM	B4 VLIM	B3 RCE	B2 COM1	B1 RCC	B0 RC1
	–	–	Stroke limit reached	Speed limit reached	Fault switching records	Control mode acknowledgement	All next records switched to	1. Switched to next record
Primary actual value Bytes 5 to 8	Depending on the parametrisation: Actual position or actual force according to the FHPP setting (PNU 523:04 or 523:08) in the set system of units (section B.1).							

## CMA<sub>X</sub>\_FB\_CTRL Record Select Mode

The below chart provides an example on how to logically perform moves in this mode of operation from the PLC/external higher order controller.

User action	Acknowledgement
PLC Selects <b>Record Select</b> Mode (See Note #1): CCON_OPM1_OperationMode = 0, CCON_OPM2_OperationMode = 0	<b>R_e_c_o_r_d_M_o_d_e_A_c_t_i_v_e = 1</b> SCON_OPM1_OperationState = 0, SCON_OPM2_OperationState = 0 (See Note #2)
PLC Selects Desired Record #: RecordNumReq = 1	No acknowledgement
PLC requests Task Start: CPOS_StartTask = 0 → 1	SPOS_AcknowledgeStart = 1, SPOS_MotionComplete = 0 (depending on the starting position), SPOS_AxisIsMoving = 1 (depending on the starting position), RecordNumStatus = ... (record currently being executed, RecordMode_PrimaryValue = ... (depending on the configuration of the I/O data)
PLC resets Task Start request: CPOS_StartTask = 0	After completing positioning: SPOS_AcknowledgeStart = 0, SPOS_AxisIsMoving = 0, SPOS_MotionComplete = 1,
Note#1) CDIR_ForceCtrlEnable = –, CDIR_VelocityCtrlAutoEnable = –	Note#2) SDIR_ForceCtrl_State = pressure/force (depending on the record), SDIR_VelocityCtrlAuto_State = 0 The status changes when there is a rising edge at the "CPOS_StartTask" input.

# CMAX\_FB\_CTRL Direct Mode

In this mode, positioning task data is formulated/stored in the PLC then sent via the output data and executed.

Data	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Output data	CCON	CPOS	CDIR	Secondary setpoint	Primary setpoint (position, force)			
Input data	SCON	SPOS	SDIR	Secondary actual value	Primary actual value (actual position, force)			

## Assignment of the control bytes (direct mode)

CCON Byte 1	B7 OPM2	B6 OPM1	B5 LOCK	B4 —	B3 RESET	B2 BRAKE	B1 STOP	B0 ENABLE
	Mode selection		Block software access	—	Reset fault	Release brake	Stop	Enable drive
CPOS Byte 2	B7 —	B6 CLEAR	B5 TEACH	B4 JOGN	B3 JOGP	B2 HOM	B1 START	B0 HALT
	—	—	Teach value	Jog negative	Jog positive	Start homing	Start positioning task	—
CDIR Byte 3	B7 —	B6 FAST	B5 XLIM	B4 VLIM	B3 CONT	B2 COM2	B1 COM1	B0 ABS
	—	Exact stop / fast stop	Deactivate stroke limit	Velocity limit reached	Tracking mode	Control mode 2 (profile)	Control mode 1 (position, force)	Absolute / relative
Secondary setpoint Byte 4	Setpoint, depending on control mode and FHPP setting (PNU 523) — Speed as percentage of default value (PNU 540) — Force ramp as percentage of default value (PNU 550) — Workpiece mass as percentage of default value (PNU 544 or PNU 551) Value range 0 to 100, no sign given. Impermissible setpoint values are limited. For the "Workpiece mass" secondary setpoint, 100% of the basic value is always used for the speed or force ramp. The setpoint value is transferred to CPOS.START with a positive edge.							
Primary setpoint Bytes 5 to 8	Setpoint value of the position or force in the set system of units (section B.1). The setpoint value is transferred to CPOS.START with a positive edge. During tracking mode, the setpoint position is transferred continuously after the start until tracking mode is ended.							

## Assignment of the status bytes (direct mode)

SCON Byte 1	B7 OPM2	B6 OPM1	B5 FCT_MMI	B4 24VL	B3 FAULT	B2 WARN	B1 OPEN	B0 ENABLED
	Operating mode acknowledgement		Software device control	Load voltage applied	Fault	Warning	Operation enabled	Drive enabled
SPOS Byte 2	B7 REF	B6 STILL	B5 DEV	B4 MOV	B3 TEACH	B2 MC	B1 ACK	B0 HALT
	Drive referenced	Standstill warning	Following error	Axis is moving	Acknowledge Teach	Motion Complete	Acknowledge Start	—
SDIR Byte 3	B7 —	B6 FAST <sup>1)</sup>	B5 XLIM	B4 VLIM	B3 CONT	B2 COM2	B1 COM1	B0 ABS <sup>1)</sup>
	—	Exact stop / fast stop active	Stroke limit reached	Speed limit reached	Tracking mode	Acknowledge control mode 2	Acknowledge control mode 1	Absolute / relative
Secondary actual value Byte 4	Speed actual value as a percentage of the default value (PNU 540). The secondary speed actual value has a sign, so positive and negative values can be displayed. The entire value range is utilised, i.e. the displayed speed lies in the range from -128 % to +127 %. Higher speeds are limited to -128 % or +127 %.							
Primary actual value Bytes 5 to 8	Actual position or force value in the set system of units (section B.1), depending on the operating status and the FHPP setting (PNU 523).							

<sup>1)</sup> The status bit only changes when the job is transferred (starting edge). All other status bits in the SDIR and RSB are updated cyclically.

# CMAX\_FB\_CTRL Direct Mode Position Ctrl – Free Profile

The below chart provides an example on how to logically perform moves in this mode of operation from the PLC/external higher order controller.

User action	Acknowledgement
PLC Selects <b>Direct Mode</b> : CCON_OPM1_OperationMode = 1, CCON_OPM2_OperationMode = 0	<b>D_i_r_e_c_t_M_o_d_e_A_c_t_i_v_e = 1</b> SCON_OPM1_OperationState = 1, SCON_OPM2_OperationState = 0
PLC Selects Desired Speed Profile Control Mode (CDIR B2): Free(Via User)=0, Auto(Via Identification Determined Values)=1, CDIR_ForceCtrlEnable = 0, <b>CDIR_VelocityCtrlAutoEnable = 0</b>	No acknowledgement <b>NOTE:</b> The configured control type is only transferred after the movement has started
DirectMode_SecondarySetPoint = xxx (Depends on FCT(Festo Configuration Tool Software) configuration of I/O data (see picture below - [Option of Velocity in % or Work Piece Mass in %])	No acknowledgement
DirectMode_PrimarySetpoint = ... (position)	No acknowledgement
PLC requests Task Start: CPOS_StartTask = 0 → 1	SDIR_ForceCtrl_State = 0, <b>SDIR_VelocityCtrlAuto_State = 0</b> , SPOS_AcknowledgeStart = 1, SPOS_MotionComplete = 0 (depending on the starting position), SPOS_AxisMoving = 1 (depending on the starting position), DirectMode_SecondaryValue = ... (Velocity %), DirectMode_PrimaryValue = ... (Depends on FCT (Festo Configuration Tool Software) configuration of I/O data (see picture below – [Option of Position or Force Data])
PLC resets Task Start request: CPOS_StartTask = 0	After completing positioning: SPOS_AcknowledgeStart = 0, SPOS_AxisMoving = 0, SPOS_MotionComplete = 1

**NOTE:** Multiple Direct Mode type moves are commonly required in an application, so for convenience, the CMAX\_FB\_MOVE\_MAM Function Block (AOI) instruction was created. This AOI instruction is used specifically to execute moves and is discussed elsewhere in this document.

The screenshot displays the Festo CMAX software interface. On the left, the 'Components' tree shows the 'CMAX: CMAX1' block with various sub-components like 'Controller', 'Axis: 1', 'Configuration', 'Application data', 'Measure/Limits', 'Position set table', 'Jog mode', 'Direct mode', 'Controller data', 'Diagnosis/IO-data', 'Trace configuration', and 'Trace data'. The 'Diagnosis/IO-data' block is highlighted. The main window shows the 'IO-data' tab, which is divided into 'Nominal values' and 'Actual values' sections. The 'Nominal values' section has a table with columns for Byte 1 through Byte 8. The 'Position' row shows 'Control bytes' for Byte 1-3 and 'Secondary nominal value' for Byte 4. The 'Force' row shows 'Control bytes' for Byte 1-3 and 'ForceRamp in %' for Byte 4. The 'Actual values' section has a similar table. The 'Position' row shows 'Status bytes' for Byte 1-3 and 'Secondary actual value' for Byte 4. The 'Force' row shows 'Status bytes' for Byte 1-3 and 'Velocity in %' for Byte 4. Red arrows point to specific fields: 'DirectMode\_SecondarySetPoint' points to the 'Secondary nominal value' field (Byte 4), 'DirectMode\_SecondaryValue' points to the 'Secondary actual value' field (Byte 4), and 'DirectMode\_PrimaryValue' points to the 'Primary actual value' field (Byte 5-8).

# CMAX\_FB\_CTRL Direct Mode Position Ctrl – Auto Profile

The below chart provides an example on how to logically perform moves in this mode of operation from the PLC/external higher order controller.

User action	Acknowledgement
PLC Selects <b>Direct Mode</b> : CCON_OPM1_OperationMode = 1, CCON_OPM2_OperationMode = 0	<b>D_i_r_e_c_t_M_o_d_e_A_c_t_i_v_e=1</b> SCON_OPM1_OperationState = 1, SCON_OPM2_OperationState = 0
PLC Selects Desired Speed Profile Control Mode (CDIR B2): Free(Via User)=0, Auto(Via Identification Determined Values)=1, CDIR_ForceCtrlEnable = 0, <b>CDIR_VelocityCtrlAutoEnable = 1</b>	No acknowledgement <b>NOTE:</b> The configured control type is only transferred after the movement has started
DirectMode_SecondarySetPoint = 0 <b>(INVALID</b> Due to use of CDIR_VelocityCtrlAutoEnable)	No acknowledgement
DirectMode_PrimarySetpoint = ... (position)	No acknowledgement
PLC requests Task Start: CPOS_StartTask = 0 → 1	SDIR_ForceCtrl_State = 0, <b>SDIR_VelocityCtrlAuto_State = 1</b> , SPOS_AcknowledgeStart = 1, SPOS_MotionComplete = 0 (depending on the starting position), SPOS_AxisIsMoving = 1 (depending on the starting position), DirectMode_SecondaryValue = ... (Velocity %), DirectMode_PrimaryValue = ... (Depends on FCT (Festo Configuration Tool Software) configuration of I/O data (see picture below – [Option of Position or Force Data])
PLC resets Task Start request: CPOS_StartTask = 0	After completing positioning: SPOS_AcknowledgeStart = 0, SPOS_AxisIsMoving = 0, SPOS_MotionComplete = 1

**NOTE:** Multiple Direct Mode type moves are commonly required in an application, so for convenience, the CMAX\_FB\_MOVE\_MAM Function Block (AOI) instruction was created. This AOI instruction is used specifically to execute moves and is discussed elsewhere in this document.

The screenshot shows the Festo CMAX software interface. On the left, the 'Components' tree is expanded to show 'CMAX: CMAX1' with sub-items: 'Controller', 'Axis: 1', 'Configuration', 'Application data', 'Measure/Limits', 'Position set table', 'Jog mode', 'Direct mode', 'Controller data', 'Diagnosis/IO-data' (highlighted), and 'Trace data'. The main window displays the 'Diagnosis memory | Reaction due to faults | IO-data' tab. It is divided into 'Nominal values' and 'Actual values' sections. The 'Nominal values' section has a table with columns for Byte 1 through Byte 8. Red arrows point to 'Byte 4' (Secondary nominal value: Velocity in %) and 'Byte 8' (Primary nominal value: Nominal position). The 'Actual values' section has a similar table. Red arrows point to 'Byte 4' (Secondary actual value: Velocity in %) and 'Byte 8' (Primary actual value: Actual position). The 'DirectMode\_SecondarySetPoint' label is at the top right, and 'DirectMode\_SecondaryValue' and 'DirectMode\_PrimaryValue' labels are at the bottom right.

# CMAX\_FB\_CTRL Direct Mode Pressure/Force Control

The below chart provides an example on how to logically perform moves in this mode of operation from the PLC/external higher order controller.

User action	Acknowledgement
PLC Selects <b>Direct Mode</b> : CCON_OPM1_OperationMode = 1, CCON_OPM2_OperationMode = 0	<b>D_i_r_e_c_t_M_o_d_e_A_c_t_i_v_e = 1</b> SCON_OPM1_OperationState = 1, SCON_OPM2_OperationState = 0
PLC Selects Force Mode Control Mode (CDIR B1): Force Select=1, Position Select=0, CDIR_VelocityCtrlAutoEnable = 0, <b>CDIR_ForceCtrlEnable = 1</b>	No acknowledgement <b>NOTE:</b> The configured control type is only transferred after the movement has started
DirectMode_SecondarySetPoint = xxx (Depends on FCT (Festo Configuration Tool Software) configuration of I/O data (see picture below - [Option of Force Ramp in % or Work Piece Mass in %])	No acknowledgement
DirectMode_PrimarySetpoint = ... (Force)	No acknowledgement
PLC requests Task Start: CPOS_StartTask = 0 → 1	SDIR_VelocityCtrlAuto_State = 0, <b>SDIR_ForceCtrl_State = 1</b> , SPOS_AcknowledgeStart = 1, SPOS_MotionComplete = 0 (depending on the starting position), SPOS_AxisIsMoving = 1 (depending on the starting position), DirectMode_SecondaryValue = ... (Velocity %), DirectMode_PrimaryValue = ... (Depends on FCT (Festo Configuration Tool Software) configuration of I/O data (see picture below – [Option of Position or Force Data])
PLC resets Task Start request: CPOS_StartTask = 0	After reaching the torque specification SPOS_AcknowledgeStart = 0, SPOS_AxisIsMoving = 0, SPOS_MotionComplete = 1 SPOS_StandstillWarning = 1 (If Axis move after Motion complete)

**NOTE:** Multiple Direct Mode type moves are commonly required in an application, so for convenience, the CMAX\_FB\_MOVE\_FAM Function Block (AOI) instruction was created. This AOI instruction is used specifically to execute moves and is discussed elsewhere in this document.

The screenshot displays the 'IO-data' configuration window for the CMAX controller. It is divided into 'Nominal values' and 'Actual values' sections. In the 'Nominal values' section, Byte 4 is configured as 'Secondary nominal value' with 'Velocity in %' selected, and Byte 5 is configured as 'Primary nominal value' with 'ForceRamp in %' selected. In the 'Actual values' section, Byte 4 is configured as 'Secondary actual value' with 'Velocity in %' selected, and Byte 5 is configured as 'Primary actual value' with 'Actual force' selected. Red arrows highlight these specific configurations.



# CMAX\_FB\_CTRL Parameterization Mode

In this mode, parameters can be transferred in the cyclical I/O data of the FHPP (Festo Handling Positioning Profile).

Data	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Output data	CCON	Subindex	Task identifier + parameter number		Parameter value			
Input data	SCON	Subindex	Reply identifier + parameter number		Parameter value			

## Assignment of the control bytes (parametrisation)

CCON Byte 1	B7 OPM2	B6 OPM1	B5 LOCK	B4 –	B3 RESET	B2 BRAKE	B1 STOP	B0 ENABLE
	Mode selection		Block software access	–	Reset fault	Release brake	Stop	Enable drive
Subindex Byte 2	Subindex of the parameter to be transferred							
Param. identifier Bytes 3+4	Job identifier and parameter number: Bit Content Description 0...11 PNU Parameter number of the parameter to be transferred 12 to 15 ReqID Job identifier, e.g. reading, writing, see section 6.1.1							
Param. value Bytes 5 to 8	Value of the parameter to be transferred (32-bit number)							

## Assignment of the status bytes (parametrisation)

SCON Byte 1	B7 OPM2	B6 OPM1	B5 FCT_MMI	B4 24VL	B3 FAULT	B2 WARN	B1 OPEN	B0 ENABLED
	Operating mode acknowledgement		Software device control	Load voltage applied	Fault	Warning	Operation enabled	Drive enabled
Subindex Byte 2	Subindex of the transferred parameter							
Param. identifier Bytes 3+4	Reply identifier and parameter number: Bit Content Description 0...11 PNU Parameter number of the parameter to be transferred 12 to 15 ResID Reply identifier, see section 6.1.1							
Param. value Bytes 5 to 8	Value of the parameter to be transferred (32-bit number)							

FPC		Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
FPC in the cyclic I/O data	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Output data	CCON	Subindex	Parameter identifier		Parameter value			
		Request byte 1 ... 7 of the FPC						
Input data	SCON	Subindex	Parameter identifier		Parameter value			
		Response byte 1 ... 7 of the FPC						

## CMAX\_FB\_CTRL Parameterization - READ

The below chart provides an example how to logically perform a Parameter **Read** in this mode from the PLC/external higher order controller.

User action	Acknowledgement
CCON_Stop = 0	SCON_OperationEnabled = 0
PLC Selects <b>Parameterisation Mode</b> : CCON_OPM1_OperationMode = 1, CCON_OPM2_OperationMode = 1	<b>P_a_r_a_m_e_t_e_r_M_o_d_e_A_c_t_i_v_e=1</b> SCON_OPM1_OperationState = 1, SCON_OPM2_OperationState = 1
PLC Selects <b>Read</b> Type parameter: <b>Req_ID = 6</b>	No acknowledgement
PLC Selects PNU number to Read (Current Error Code on CMAX LED Display): Req_PNU = 224	No acknowledgement
PLC Selects SubIndex (Current Error Code on CMAX LED Display): Req_SubIndex = 1	No acknowledgement
PLC sends task request: Req_Enable = 0 → 1	Req_Complete = 1, Req_Failed = 0 (No Problem with transmission), Res_Req_PNU = 224, Res_SubIndex = 1, Res_ID = 5 (successful transmission), Res_Value = 50 (No/Low Air Pressure), Res_FaultCode = ?? (0=No Errors in Transmission of Parameter)
PLC resets task request: Req_Enable = 0	Req_Complete = 0, Req_Failed = 0 (Response Area Cleared), Res_Req_PNU = 0 (Response Area Cleared), Res_SubIndex = 0 (Response Area Cleared), Res_ID = 0 (Response Area Cleared), Res_Value = 0 (Response Area Cleared), Res_FaultCode = 0 (Response Area Cleared)

**NOTE:** For additional convenience and multiple FPC (Festo Parameter Channel) calls, the CMAX\_FB\_FPC Function Block (AOI) instruction was created. This AOI instruction **retains** previously sent **data** after the Req\_Enable has been reset. This instruction is discussed elsewhere in this document.

## CMAX\_FB\_CTRL Parameterization - WRITE

The below chart provides an example how to logically perform a Parameter **Write** in this mode from the PLC/external higher order controller.

User action	Acknowledgement
CCON_Stop = 0	SCON_OperationEnabled = 0
PLC Selects <b>Parameterisation Mode</b> : CCON_OPM1_OperationMode = 1, CCON_OPM2_OperationMode = 1	<b>P_a_r_a_m_e_t_e_r_M_o_d_e_A_c_t_i_v_e=1</b> SCON_OPM1_OperationState = 1, SCON_OPM2_OperationState = 1
PLC Selects <b>Write</b> Type parameter: <b>Req_ID = 8</b>	No acknowledgement
PLC Selects PNU number to Write (record List Set point): Req_PNU = 404	No acknowledgement
PLC Selects SubIndex (Record #2) Req_SubIndex = 2	No acknowledgement
PLC sets Write Data for the PNU: Req_Value = 13456 (Set point)	No acknowledgement
PLC sends task request: Req_Enable = 0 → 1	Req_Complete = 1, Req_Failed = 0 (No Problem with transmission), Res_Req_PNU = 404, Res_SubIndex = 2, Res_ID = 5 (successful transmission), Res_Value = 13456 (should Mirror of data sent), Res_FaultCode = ?? (0=No Errors in Transmission of Parameter)
PLC resets task request: Req_Enable = 0	Req_Complete = 0, Req_Failed = 0 (Response Area Cleared), Res_Req_PNU = 0 (Response Area Cleared), Res_SubIndex = 0 (Response Area Cleared), Res_ID = 0 (Response Area Cleared), Res_Value = 0 (Response Area Cleared), Res_FaultCode = 0 (Response Area Cleared)

**NOTE:** For additional convenience and multiple FPC (Festo Parameter Channel) calls, the CMAX\_FB\_FPC Function Block (AOI) instruction was created. This AOI instruction **retains** previously sent **data** after the Req\_Enable has been reset. This instruction is discussed elsewhere in this document.

# CMAX\_FB\_CTRL Commissioning Mode

Commissioning operating mode is for putting the CMAX into operation, like executing such functions like jogging, teaching, referencing Identification, movement test, and other commissioning functions.

Data	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Output data	CCON	CPOS	Function	Param. 1	Parameter 2 (e.g. current workpiece mass)			
Input data	SCON	SPOS	Function	Progress	Primary actual value (actual position)			

## Assignment of the control bytes (commissioning mode)

CCON Byte 1	B7 OPM2	B6 OPM1	B5 LOCK	B4 —	B3 RESET	B2 BRAKE	B1 STOP	B0 ENABLE
	Mode selection		Block soft-ware access	—	Reset fault	Release brake	Stop	Enable drive
CPOS Byte 2	B7 —	B6 CLEAR	B5 TEACH	B4 JOGN	B3 JOGP	B2 HOM	B1 START	B0 HALT
	—	—	Teach value	Jog negative	Jog positive	Start homing	Start positioning task	—
Function Byte 3	The function number selects the starting commissioning operation in commissioning mode. The value is interpreted as enumeration. The functions are executed with a rising edge at CPOS.START.							
	Value	Function	Description	Param. 1	Param. 2			
	0	Reserved	Not permissible	—	—			
	1	Identification	Execute identification run	= 0	Workpiece mass			
	2	Movement test	Execute movement test	= 0	= 0			
	3 to 255	Reserved	Not permissible	—	—			
	When reserved functions are executed, the CMAX signals a corresponding error.							
Param. 1 Byte 4	When executing a commissioning operation: Reserved = 0 When teaching: Teach target, see section 3.2.7. A zero (0=) must be transferred in setpoint value bytes which aren't used.							
Param. 2 Bytes 5 to 8	Only for "Identification" commissioning operation: Current workpiece mass in the set system of units (see section B.1). A zero (0=) must be transferred in setpoint value bytes which aren't used.							

## Assignment of the status bytes (commissioning mode)

SCON Byte 1	B7 OPM2	B6 OPM1	B5 FCT_MMI	B4 24VL	B3 FAULT	B2 WARN	B1 OPEN	B0 ENABLED
	Operating mode acknowledgement		Software device control	Load voltage applied	Fault	Warning	Operation enabled	Drive enabled
SPOS Byte 2	B7 REF	B6 STILL	B5 DEV	B4 MOV	B3 TEACH	B2 MC	B1 ACK	B0 HALT
	Drive referenced	Standstill warning	Following error	Axis is moving	Acknowledge Teach	Motion Complete	Acknowledge Start	—
Function Byte 3	Acknowledgement of the commissioning operation currently being executed.							
Progress Byte 4	When executing a commissioning operation: The progress display in the status data shows to what degree the function has progressed for long operations. Display in percent (0% to 100%). There could be jumps in the progress display (e. g. from 24% to 60%). At the end of the function, the progress counter is set to 255 (0xFF). When teaching: Teach target, see section 3.2.7.							
Primary actual value Bytes 5 to 8	Depending on the parametrisation: Actual position or actual force according to the FHPP setting (PNU 523:04 or 523:08) in the set system of units (section B.1).							

## CMAX\_FB\_CTRL Commissioning

The below chart provides an example how to logically perform general functions in this mode from the PLC/external higher order controller.

User action	Acknowledgement
CCON_Stop = 0	SCON_OperationEnabled = 0
PLC Selects <b>Commissioning Mode</b> : CCON_OPM1_OperationMode = 0, CCON_OPM2_OperationMode = 1	<b>C_o_m_m_i_s_i_o_n_M_o_d_e_A_c_t_i_v_e = 1</b> SCON_OPM1_OperationState = 0, SCON_OPM2_OperationState = 1
PLC Selects the Function Type to perform: <b>ComMode_Function = xxx</b>	No acknowledgement
PLC sends task request: CPOS_StartTask = 0 → 1	SPOS_AcknowledgeStart = 1, SPOS_MotionComplete = 0, SPOS_AxisIsMoving = 1, DirectMode_PrimaryValue = ... (current position), ActualComMode_Function = ... (current function), ComMode_Progress = ... (progress)
PLC resets task request: CPOS_StartTask = 0	After completing the commissioning operation: SPOS_AxisIsMoving = 0, SPOS_MotionComplete = 1

## CMAX\_FB\_FPC Festo Parameter Channel Function Block (AOI)

The FPC is used for transmitting parameters in which the process takes several bus cycles.

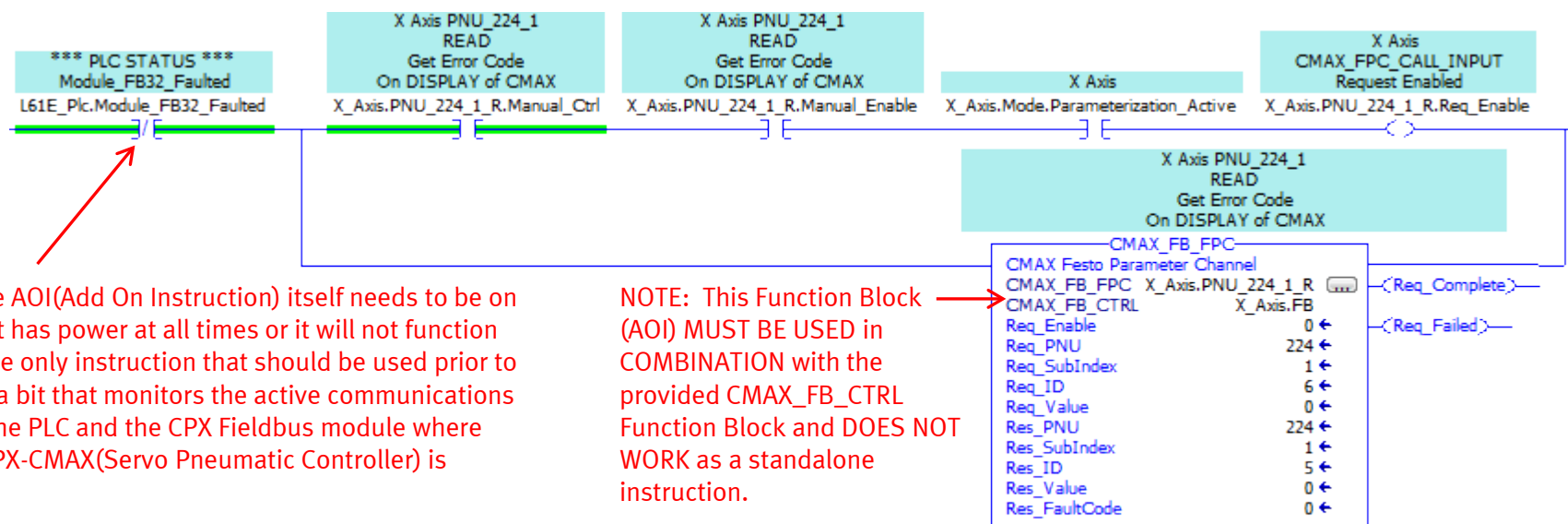
In Parameterisation mode the CMAX\_FB\_FPC can be used to **transmit one parameter at a time** in the cyclic I/O data. The PLC enters the request in the output data and waits until the CMAX has entered a response in the input data.

The PLC sends a **request** to the CPX-CMAX consisting of the following data:

- ☐ parameter number (Req\_PNU)
- ☐ sub-index (Req\_SubIndex)
- ☐ request/task identifier to determine the call type (Req\_ID)
- ☐ when performing a write - value (Req\_Value)

The CPX-CMAX sends a **response** with the following data:

- ☐ PNU number(Res\_PNU)
- ☐ sub-index (Res\_SubIndex)
- ☐ response identifier (Res\_ID)
- ☐ value of response of Read or Write (Res\_Value)
- ☐ fault (if present) (Res\_FaultCode)



NOTE: The AOI(Add On Instruction) itself needs to be on a rung that has power at all times or it will not function correct. The only instruction that should be used prior to the AOI is a bit that monitors the active communications between the PLC and the CPX Fieldbus module where the CPX-CPX-CMAX(Servo Pneumatic Controller) is installed.

NOTE: This Function Block (AOI) MUST BE USED in COMBINATION with the provided CMAX\_FB\_CTRL Function Block and DOES NOT WORK as a standalone instruction.

# CMAX\_FB\_FPC Description of Parameters

This Function Block **MUST BE USED in COMBINATION** with the provided CMAX\_FB\_CTRL function block and **DOES NOT WORK** as a standalone instruction. The parameters described below are identical to the ones described earlier in the CMAX\_FB\_CTRL Function Block. The advantage of the CMAX\_FB\_FPC function block is the ability to pre-program PNU data and to retain information from these requests.

Parameter Mode Active		
FHPP Manual	Input / output	Description
	Req_Enable	The transmission of a parameter is started by a <b>rising edge</b>
	Req_Complete	= 0: Transmission was not initiated = 1: Transmission was Completed
	Req_Failed	= 0: No error while transmitting a parameter = 1: Error transmitting a parameter
BYTE 3-4	Param. Identifier ReqID (Bits12-15)	= 6: Read parameters = 8: Write parameters
	Param. Identifier PNU (Bits0-11)	Parameter number of the corresponding parameter
BYTE 3	Subindex	Subindex of the corresponding parameter
BYTE 5-8	Param. Value	Parameter value when wRITING a parameter
BYTE 3-4	Param. Identifier ResID (Bits12-15)	= 6: Read parameters = 8: Write parameters
	Param. Identifier PNU (Bits0-11)	Display of the current parameter number
BYTE 3	Subindex	Display of the current subindex
BYTE 5-8	Param. Value	Response Value = 0: No Reply = 5: Parameter Value was transferred = 7: Error/Parameter request Failed
	Res_FaultCode (See Note 2)	Parameter Transmission Errors

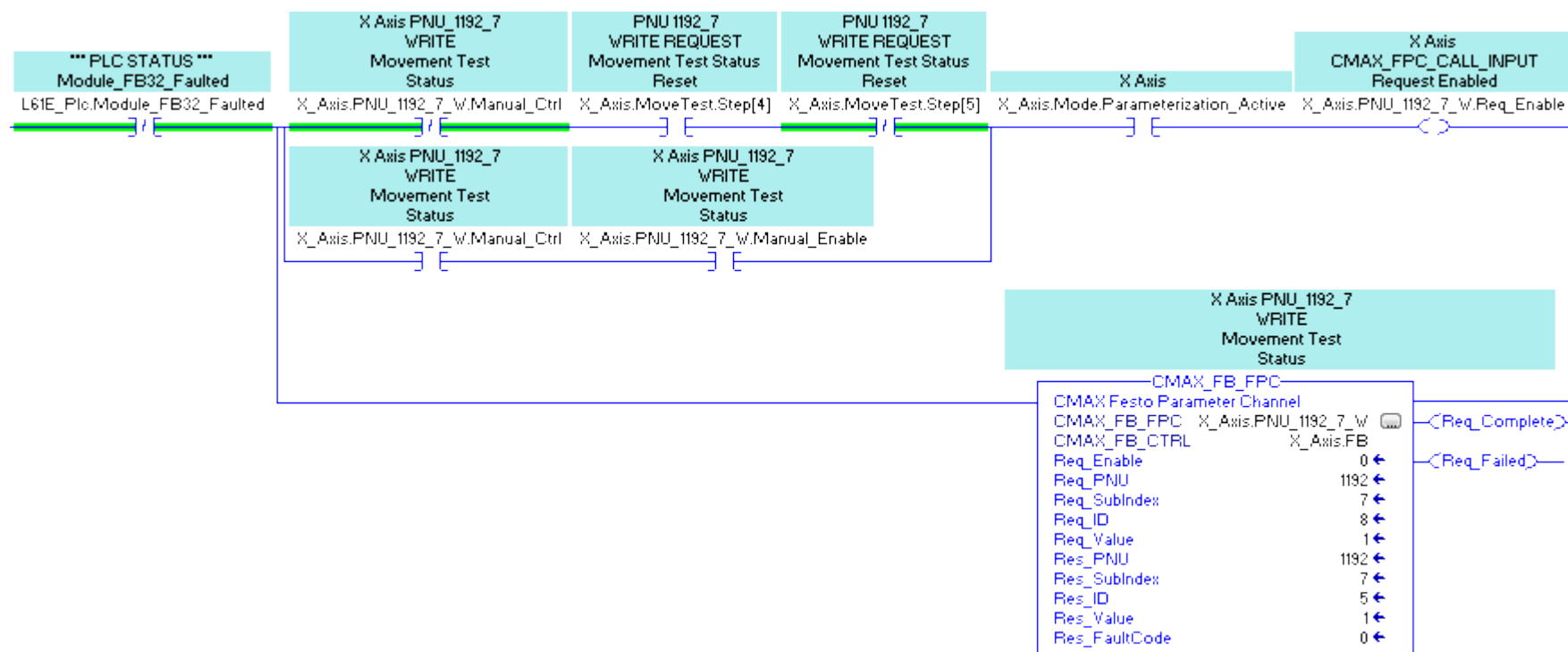
Parameter Transmission Errors (Res_FaultCode)	
Error	Error description
0	Impermissible PNU.
1	Parameter value cannot be changed.
2	Lower or upper value limit exceeded.
3	Invalid sub index.
11	No higher-order controller. FCT must accept device control in order to write this parameter. This error can only be generated via the service interface.
12	The password entered is wrong.
17	Request cannot be carried out due to operating status. Please check operating mode, stop and enable signals.
101	Request ID is not supported.
102	Parameter cannot be read (password).
103	The system of measurement has not been configured yet. Access to the parameter is not possible.
104	The cylinder type has not been configured yet. Access to the parameter is not possible.
105	The system of measurement has already been configured and cannot be changed without a data reset.
106	Cylinder type cannot be changed since it does not match the system of measurement.
107	The value cannot be changed since there is identification data. Please reset the identification data before writing the value.
108	The parameter value does not match the hardware recognised. (Note: the cylinder type must match the measuring system)
109	Serial numbers cannot be changed before the identification data has been reset.

# CMAF\_FB\_FPC Example Code

The CMAF\_FB\_FPC **transmits one parameter at a time**. The external PLC code structure operating on the CMAF\_FB\_FPC should be as shown below.

Prior to setting the Req\_Enable, the PLC needs to verify the following:

- ❑ communication to the CPX-CMAF is operational (L61E\_Plc.Module\_FB32\_Faulted)
- ❑ parameterisation mode is active (Mode.Parameterization\_Active).
- ❑ correct data is populated in such parameters as Req\_PNU, Req\_SubIndex, Req\_ID, and Req\_Value.





## CMAX\_FB\_FPC Example Code

The below chart provides an example how to logically perform a Parameter **Write** from the PLC/external higher order controller.

User action	Acknowledgement
CCON_Stop = 0	SCON_OperationEnabled = 0
PLC Selects <b>Parameterisation Mode</b> : CCON_OPM1_OperationMode = 1, CCON_OPM2_OperationMode = 1	<b>P_a_r_a_m_e_t_e_r_M_o_d_e_A_c_t_i_v_e=1</b> SCON_OPM1_OperationState = 1, SCON_OPM2_OperationState = 1
PLC Selects PNU number (Commissioning Function Movement Test Status) Req_PNU = 1192	No acknowledgement
PLC Selects SubIndex (Commissioning Function Movement Test Status) Req_SubIndex = 7	No acknowledgement
PLC Selects <b>Write</b> Type parameter: <b>Req_ID = 8</b>	No acknowledgement
PLC sets Write Data for the PNU: Define status of the movement test. = 1: Movement test is reset and must be carried out again = 2: Movement test is set and doesn't have to be carried out and is skipped Req_Value = 1	No acknowledgement
PLC sends task request: Req_Enable = 0 → 1	Req_Complete = 1, Req_Failed = 0 (No Problem with transmission, otherwise see previous page for fault codes), Res_Req_PNU = 1192 (should Mirror of data sent), Res_SubIndex = 7 (should Mirror of data sent), Res_ID = 5 (successful transmission), Res_Value = 1 (response of parameter change), Res_FaultCode = ?? (0=No Errors in Transmission of Parameter, otherwise see previous page for fault codes)
PLC resets task request: Req_Enable = 0	Req_Complete = 0, Req_Failed = 0 (No Problem with transmission), Res_Req_PNU = 1192 ( <b>Retains last Request Data</b> ), Res_SubIndex = 7 ( <b>Retains last Request Data</b> ), Res_ID = 5 ( <b>Retains last Request Data</b> ), Res_Value = 1 ( <b>Retains last Request Data</b> ), Res_FaultCode = 0 ( <b>Retains last Request Data</b> )

## CMAX\_FB\_MOVE\_PAM Position Axis Move Function Block (AOI)

This instruction executes a **POSITION** Axis Move while in Direct Mode. Only **one instruction of this type can be enabled a time**. The PLC enters the request in the output data and waits until the CMAX has entered a response in the input data.

The PLC sends a **request** to the CPX-CMAX consisting of the following data:

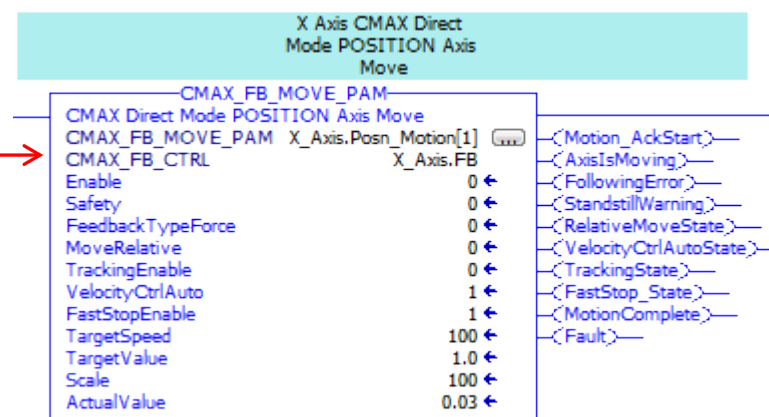
- ☐ pre/during move safety (Safety)
- ☐ feedback type expected and set in the FCT (FeedbackTypeForce)
- ☐ relative/absolute move required (MoveRelative)
- ☐ continuous tracking mode (Tracking Enable)
- ☐ velocity mode profile auto/free (VelocityCtrlAuto)
- ☐ fast/exact stop function (FastStopEnable)
- ☐ target speed% (TargetSpeed)
- ☐ target position (TargetValue)

**NOTE:** The AOI(Add On Instruction) itself needs to be on a rung that has power at all times or it will not function correct. The only instruction that should be used prior to the AOI is a bit that monitors the active communications between the PLC and the CPX Fieldbus module where the CPX-CPX-CMAX(Servo Pneumatic Controller) is installed.

The CPX-CMAX sends a **response** with the following data:

- ☐ actual position (ActualValue)
- ☐ motion start acknowledge (Motion\_AckStart)
- ☐ axis moving status (AxisIsMoving)
- ☐ following error status (FollowingError)
- ☐ standstill warning status (StandstillWarning)
- ☐ relative move status (RelativeMoveState)
- ☐ velocity mode profile auto/free (VelocityCtrlState)
- ☐ fast/exact stop function (FastStopState)
- ☐ motion complete status (MotionComplete)
- ☐ fault present (Fault)

**NOTE:** This Function Block (AOI) **MUST BE USED** in **COMBINATION** with the provided CMAX\_FB\_CTRL Function Block and **DOES NOT WORK** as a standalone instruction.



## CMAX\_FB\_MOVE\_PAM Description of Parameters for PLC Output

Input/Output	Type	Description
Enable	BOOL	With a rising edge the current nominal values will be transferred and positioning started.
Safety	BOOL	Pre Move and During Move condition for operation
FeedbackTypeForce	BOOL	In the FCT (Festo Configuration Tool Software) the user can configure the "ActualValue" to be either Force or Position feedback. This parameter configures the instruction to match the type you have configured for monitoring: =0: Position Feedback, =1: Force Feedback
MoveRelative	BOOL	= 0: Specified "TargetValue" is absolute (with respect to the project zero point) = 1: Specified "TargetValue" is relative to the last set-point/actual value (for MC) or to the actual value (if there is no MC).
TrackingEnable	BOOL	Activates continuous tracking mode (continuous set-point specification) = 0: Disable continuous tracking mode, = 1: Activate continuous tracking mode
VelocityCtrlAuto	BOOL	= 0: Free profile: Speed and acceleration are freely coded by the programmer (can be limited to values determined during identification) = 1: Auto profile: Speed and accelerations are specified by the controller (Speed and accelerations are selected by the controller according to the identification so that the target position is reached as quickly as possible without overshooting.)
FastStopEnable	BOOL	Control when or how the desired "TargetValue" has been reached (MC is only set when the job is completed according to the quality class) NOTE: This is ignored when Continuous tracking mode enabled so the user still requires a Velocity% specified. = 0: Exact stop, = 1: Fast stop
TargetSpeed	SINT	Secondary Set-point Value (depends FHPP setting - PNU 523 [Velocity or Work piece Mass]) Velocity: Percent(%) of the default value (PNU 540), Work piece mass: Percent (%) of the default value (PNU 544 or PNU 551). Value range 0 to 100, no sign. NOTE: Target Velocity is IGNORED when the FastStopEnable=1
TargetValue	REAL	Primary Set-point of the position in actual units (i.e. 234.52 in PLC represents 234.52mm in controller). The target value is accepted with a positive edge at Enable, and in the case of tracking mode, the target value is accepted continuously until the Enable signal is removed.
Scale	INT	The controller uses a position factor and depending on the axis type this differs. For Linear Axis (DGCI, DNCI) the exponent of "10-2" is used. For example, if the Actual position of the axis is 123.67mm, the PLC Position data would be 12367. For the DSMI, the exponent is "10-1" is used. For example, if the Actual position of the axis is 123.6°, the PLC Position data would be 1236. The embedded code inside the function block now uses the "Scale" instead of the hardcoded "100" for display of "actual value (position units)

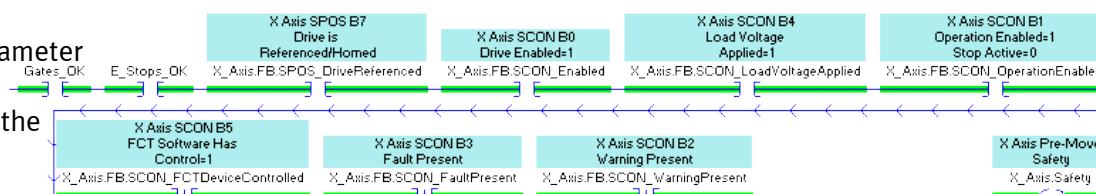
## CMAX\_FB\_MOVE\_PAM Description of Parameters for PLC Input

Input/Output	Type	Description
AcknowledgeStart	BOOL	= 0: Ready for start = 1: Start movement executed
AxisIsMoving	BOOL	Movement monitoring = 0: Drive is not moving (axis speed signal < limit) = 1: Drive is moving
FollowingError	BOOL	Following error or tolerance monitoring = 0: No following error / within tolerance = 1: Following error active / outside of tolerance
StandstillWarning	BOOL	= 0: Standstill warning Inactive = 1: Standstill warning active, drive moved after MC (position control)g not active
RelativeMoveState	BOOL	= 0: Specified "TargetValue" is absolute (with respect to the project zero point) = 1: Specified "TargetValue" is relative to the last set-point/actual value (for MC) or to the actual value (if there is no MC).
VelocityCtrlAuto_State	BOOL	= 0: Free profile: Speed and acceleration are freely coded by the programmer = 1: Auto profile: Speed and accelerations are specified by the controller determined during identification
TrackingStatus	BOOL	Continuous tracking mode (continuous set-point specification) = 0: Continuous tracking mode Disable d, = 1: Continuous tracking mode Activated
FastStop_State	BOOL	Control when the desired "TargetValue" has been reached = 0: Exact stop active, = 1: Fast stop active, malfunction is signaled
FaultPresent	BOOL	=0: No Fault =1: Fault Present
MotionComplete	BOOL	= 0: Positioning job Active = 1: Positioning job completed, where applicable with error (MC is set for the first time after switch-on [status "Drive disabled"])
ActualValue	REAL	Primary Actual Position in actual units (NOTE: Please see previous page description on Parameter "Scale"). This is also determined by the setting of the "FeedbackTypeForce" parameter. (Actual value of the position in the set system of units depending on the operating status and the FHPP setting PNU 523)

# CMAX\_FB\_MOVE\_PAM Example Code

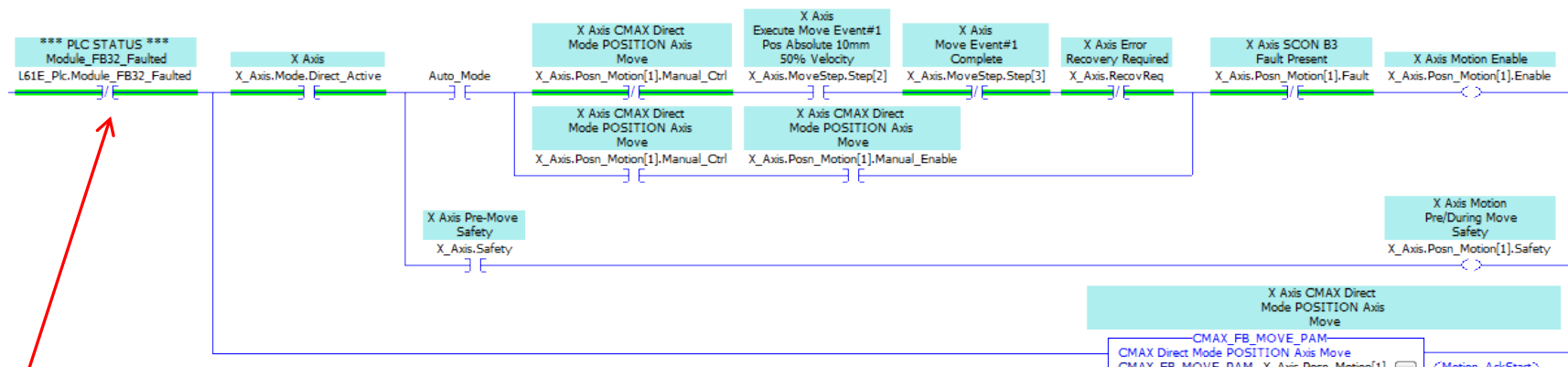
This instruction **executes one positional move at one time** and the user needs to assure that multiple calls do not happen simultaneously. The external PLC code structure operating on the CMAX\_FB\_MOVE\_PAM should be as shown below.

Before activation of the “Enable”, the “Safety” parameter needs to be true (prior to and during the move). Some common monitoring functions are shown in the safety rung here.

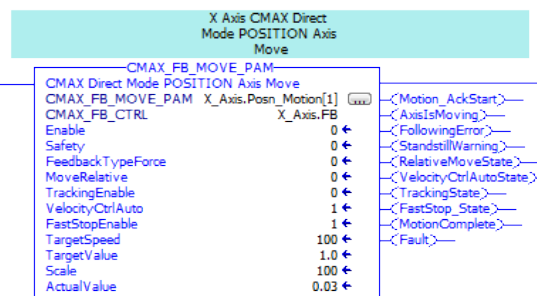


Prior to setting the “Enable”, the PLC needs to verify the following:

- ❑ communication to the CPX-CMAX is operational (L61E\_Plc.Module\_FB32\_Faulted)
- ❑ direct mode is active (Mode.Direct\_Active).
- ❑ correct data is populated in such parameters as TargetSpeed, TargetValue.



**NOTE:** The AOI(Add On Instruction) itself needs to be on a rung that has power at all times or it will not function correct. The only instruction that should be used prior to the AOI is a bit that monitors the active communications between the PLC and the CPX Fieldbus module where the CPX-CPX-CMAX(Servo Pneumatic Controller) is installed.



## CMAX\_FB\_MOVE\_PAM Example Code

The below chart provides an example on how to logically perform moves in this mode of operation from the PLC/external higher order controller.

User action	Acknowledgement
PLC Selects <b>Direct Mode</b> : CCON_OPM1_OperationMode = 1, CCON_OPM2_OperationMode = 0	<b>D_i_r_e_c_t_M_o_d_e_A_c_t_i_v_e = 1</b> SCON_OPM1_OperationState = 1, SCON_OPM2_OperationState = 0 Direct_Active = 1
PLC verifies Pre Enable functions	Manual_Ctrl = 1 or Manual_Ctrl = 0 depending on auto mode of desire to activate instruction manually by force Fault = 0 Safety = 1
PLC/User configures desired Options: Select feedback type: FeedbackTypeForce = 0 (Position Feedback) Select desired move type (CDIR B0): MoveRelative = 0 (absolute) Select tracking mode (CDIR B3): TrackingEnable = 0 (Tracking Disabled) Select velocity control type (CDIR B2): VelocityCtrlEnable = 0 (Free-Via User) Select how target is acquired (CDIR B6): FastStopEnable = 0 (Exact Stop)	No acknowledgement
TargetSpeed= 95 (95% of configured maximum speed)	No acknowledgement
TargetValue= 50.59 (Position Target is 50.59mm)	No acknowledgement
PLC requests Task Start: Enable= 0 → 1	Motion_AckStart = 1 (Motion is executing) AxisIsMoving = 1 (Drive is moving), FollowingError = 0 (no following error exists), RelativeMoveState = 0 (Absolute type move active), VelocityCtrlAuto_State = 0 (Free-Via User) TrackingState = 0 (No Tracking Activated), FastStop_State = 0 (Exact stop Active) MotionComplete = 0, ActualValue = xxx (current position)
PLC resets Task Start request: Enable= 0	After completing positioning: Motion_AckStart = 0, AxisIsMoving = 0, MotionComplete = 1 StandstillWarning = 1 (If Axis move after Motion complete)

## CMAX\_FB\_MOVE\_FAM Position Axis Move Function Block (AOI)

This instruction executes a **FORCE** Axis Move while in Direct Mode.

Only **one instruction of this type can be enabled a time**. The PLC enters the request in the output data and waits until the CMAX has entered a response in the input data.

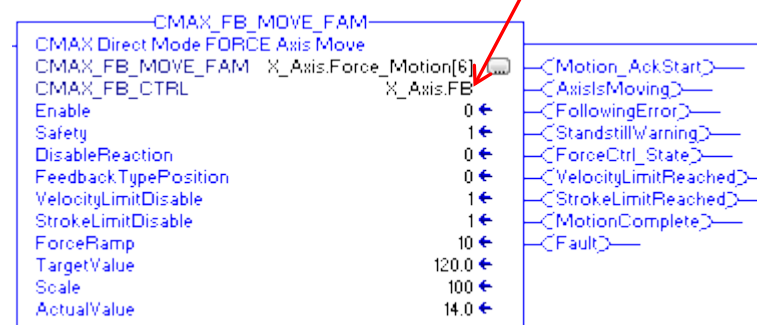
The PLC sends a **request** to the CPX-CMAX consisting of the following data:

- ☐ pre/during move safety (Safety)
- ☐ feedback type expected and set in the FCT (FeedbackTypePosition)
- ☐ velocity limit disable during movement (VelocityLimitDisable)
- ☐ stroke limit monitoring disable (StrokeLimitDisable)
- ☐ force ramp% (ForceRamp)
- ☐ target force in Newtons (TargetValue)

The CPX-CMAX sends a **response** with the following data:

- ☐ actual force or position depending on FeedbackTypePosition (ActualValue)
- ☐ motion start acknowledge (Motion\_AckStart)
- ☐ axis moving status (AxisIsMoving)
- ☐ following error status (FollowingError)
- ☐ standstill warning status (StandstillWarning)
- ☐ velocity limit achieved status (VelocityLimitReached)
- ☐ stroke limit achieved status (StrokeLimitReached)
- ☐ motion complete status (MotionComplete)
- ☐ fault present (Fault)

**NOTE:** This Function Block (AOI) **MUST BE USED** in COMBINATION with the provided CMAX\_FB\_CTRL Function Block and **DOES NOT WORK** as a standalone instruction.



## CMAX\_FB\_MOVE\_FAM Description of Parameters for PLC Output

Input/Output	Type	Description
Enable	BOOL	With a rising edge the current nominal values will be transferred and positioning started.
Safety	BOOL	Pre Move and During Move condition for operation
DisableReaction	SINT	Enable Off Reaction =0: Retains the Force that was specified in the TargetValue input parameter, =1: Apply "Zero" Force (as original design), =2: Switch to Position Mode as quickly as possible. The system immediately switches back to position control, while in standstill position the set point position is set as = actual position.
FeedbackTypePosition	BOOL	In the FCT (Festo Configuration Tool Software) the user can configure the "ActualValue" to be either Force or Position feedback. This parameter configures the instruction to match the type you have configured for monitoring: =0: Force Feedback, =1: Position Feedback
VelocityLimitDisable	BOOL	= 0: Activate speed limit = 1: Deactivate speed limit
StrokeLimitDisable	BOOL	= 0: Activate stroke limit = 1: Deactivate stroke limit
ForceRamp	SINT	Secondary Set-point Value (depends FHPP setting - PNU 523 [Force Ramp or Work piece Mass]) Force ramp: in percent of the default value (PNU 550), Work piece mass: Percent (%) of the default value (PNU 544 or PNU 551). Value range 0 to 100, no sign.
TargetValue	REAL	Primary Set-point of the Force in Newton's (i.e. 23 in PLC represents 23 Newton's controller).
Scale	INT	The controller uses a position factor and depending on the axis type this differs. For Linear Axis (DGCI, DNCI) the exponent of "10-2" is used. For example, if the Actual position of the axis is 123.67mm, the PLC Position data would be 12367. For the DSML, the exponent is "10-1" is used. For example, if the Actual position of the axis is 123.6°, the PLC Position data would be 1236. The embedded code inside the function block now uses the "Scale" instead of the hardcoded "100" for display of "actual value (position units)



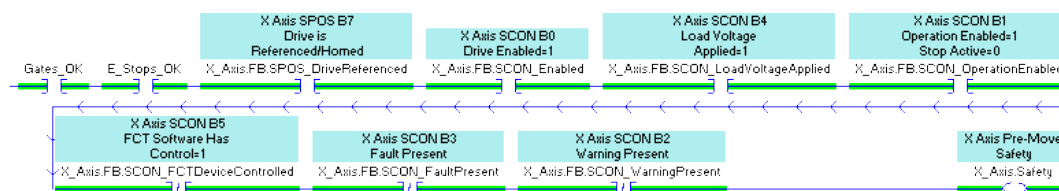
## CMA<sub>X</sub> FB\_MOVE\_FAM Description of Parameters for PLC Input

Input/Output	Type	Description
AcknowledgeStart	BOOL	= 0: Ready for start = 1: Start movement executed
AxisIsMoving	BOOL	Movement monitoring = 0: Drive is not moving (axis speed signal < limit) = 1: Drive is moving
FollowingError	BOOL	Following error or tolerance monitoring = 0: No following error / within tolerance = 1: Following error active / outside of tolerance
StandstillWarning	BOOL	= 0: Standstill warning Inactive = 1: Standstill warning active, drive moved after MC (position control)g not active
ForceCtrlState	BOOL	This is set internal to the instruction and should always be = 1 when the instruction Enable = 1 = 0: Position control active = 1: Force control active
VelocityLimitReached	BOOL	= 0: Speed limit not reached = 1: Speed limit reached
StrokeLimitReached	BOOL	= 0: Stroke limit not reached = 1: Stroke limit reached, malfunction is signaled
FaultPresent	REAL	=0: No Fault =1: Fault Present
MotionComplete	BOOL	= 0: Force job Active = 1: Target Force Achieved, (MC is set for the first time after switch-on [status "Drive disabled"])
ActualValue	REAL	Primary Actual Position in actual units . (NOTE: Please see previous page description on Parameter "Scale"). This is also determined by the setting of the "FeedbackTypePosition" parameter. (Actual value of the position in the set system of units depending on the operating status and the FHPP setting PNU 523)

# CMAx\_FB\_MOVE\_FAM Example Code

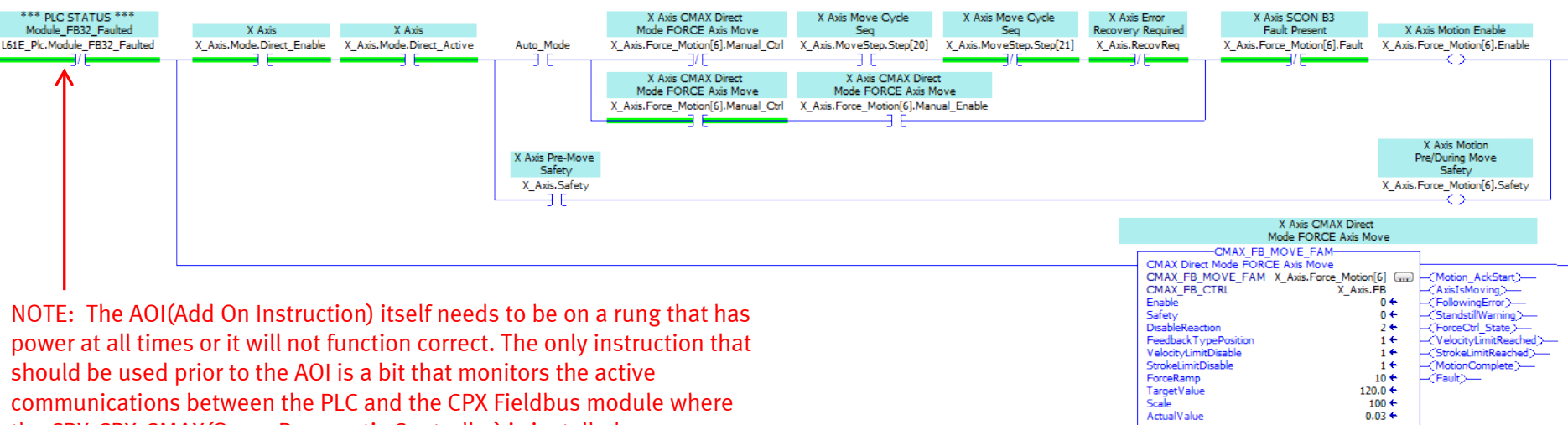
This instruction **executes one positional move at one time** and the user needs to assure that multiple calls do not happen simultaneously. The external PLC code structure operating on the CMAx\_FB\_MOVE\_FAM should be as shown below.

Before activation of the “Enable”, the “Safety” parameter needs to be true (prior to and during the move). Some common monitoring functions are shown in the safety rung here.



Prior to setting the “Enable”, the PLC needs to verify the following:

- ❑ communication to the CPX-CMAx is operational (L61E\_Plc.Module\_FB32\_Faulted)
- ❑ direct mode is active (Mode.Direct\_Active).
- ❑ correct data is populated in such parameters as ForceRamp, TargetValue.



**NOTE:** The AOI(Add On Instruction) itself needs to be on a rung that has power at all times or it will not function correct. The only instruction that should be used prior to the AOI is a bit that monitors the active communications between the PLC and the CPX Fieldbus module where the CPX-CPX-CMAx(Servo Pneumatic Controller) is installed.

## CMAX\_FB\_MOVE\_FAM Example Code

The below chart provides an example on how to logically perform moves in this mode of operation from the PLC/external higher order controller.

User action	Acknowledgement
PLC Selects <b>Direct Mode</b> : CCON_OPM1_OperationMode = 1, CCON_OPM2_OperationMode = 0	<b>D_i_r_e_c_t_M_o_d_e_A_c_t_i_v_e = 1</b> SCON_OPM1_OperationState = 1, SCON_OPM2_OperationState = 0 Direct_Active = 1
PLC verifies Pre Enable functions	Manual_Ctrl = 1 or Manual_Ctrl = 0 depending on auto mode of desire to activate instruction manually by force Fault = 0 Safety = 1
PLC/User configures desired Options: Select feedback type: FeedbackTypeForce = 0 (Position Feedback) Configure Monitoring (CDIR B4): VelocityLimitDisable = 1 (No Velocity Limit) Configure Monitoring (CDIR B5): StrokeLimitDisable = 1 (No Stroke Limit)	No acknowledgement
ForceRamp = 100 (Motion Desired Ramp (Percent(%) of Max))	No acknowledgement
TargetValue= 200.00 (Force Target is 200.00 Newton's)	No acknowledgement
PLC requests Task Start: Enable= 0 → 1	Motion_AckStart = 1 (Motion is executing) AxisIsMoving = 1 (Drive is moving), FollowingError = 0 (no following error exists), ForceCtrl_State = 1 (Force Mode Active), VelocityLimitReached = 0 (Velocity Limit was NOT Reached) StrokeLimitReached = 0 (Stroke Limit was NOT Reached) MotionComplete = 0 , ActualValue = xxx (current position)
PLC resets Task Start request: Enable= 0	After completing positioning: Motion_AckStart = 0, AxisIsMoving = 0, MotionComplete = 1 StandstillWarning = 1 (If Axis move after Motion complete)

## CMAX\_FB\_MOVE\_RAM Position Axis Move Function Block (AOI)

This instruction executes a **RECORD (Position or Force)** Axis Move while in Record Mode.

Only **one instruction of this type can be enabled a time**. The PLC enters the request in the output data and waits until the CMAX has entered a response in the input data.

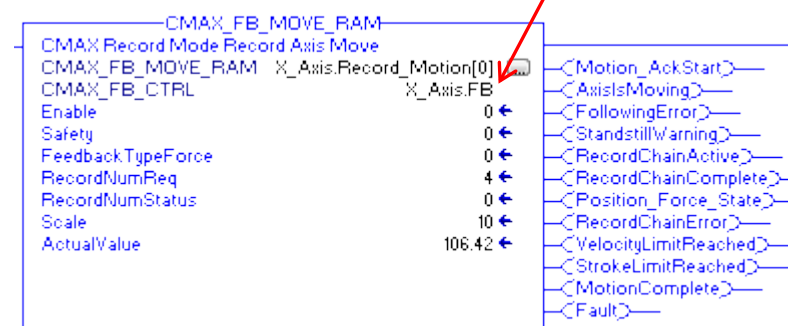
The PLC sends a **request** to the CPX-CMAX consisting of the following data:

- ☐ pre move safety (Safety)
- ☐ feedback type expected and set in the FCT (FeedbackTypeForce)
- ☐ desired record number to execute (RecordNumReq)
- ☐ actual or last record number status (RecordNumStatus)

The CPX-CMAX sends a **response** with the following data:

- ☐ actual force or position depending on FeedbackTypeForce (ActualValue)
- ☐ motion start acknowledge (Motion\_AckStart)
- ☐ axis moving status (AxisIsMoving)
- ☐ following error status (FollowingError)
- ☐ standstill warning status (StandstillWarning)
- ☐ record chaining is active (RecordChainActive)
- ☐ record chaining has complete last record (RecordChainComplete)
- ☐ position or force used in record (Position\_Force\_State)
- ☐ record chain error (RecordChainError)
- ☐ velocity limit achieved status (VelocityLimitReached)
- ☐ stroke limit achieved status (StrokeLimitReached)
- ☐ motion complete status (MotionComplete)
- ☐ fault present (Fault)

**NOTE:** This Function Block (AOI) **MUST BE USED** in COMBINATION with the provided CMAX\_FB\_CTRL Function Block and **DOES NOT WORK** as a standalone instruction.



## CMAX\_FB\_MOVE\_RAM Description of Parameters for PLC Output

Input/Output	Type	Description
Enable	BOOL	With a rising edge the current nominal values will be transferred and positioning started.
Safety	BOOL	Pre Move and During Move condition for operation
FeedbackTypeForce	BOOL	In the FCT (Festo Configuration Tool Software) the user can configure the “ActualValue” to be either Force or Position feedback. This parameter configures the instruction to match the type you have configured for monitoring: =0: Position Feedback, =1: Force Feedback
RecordNumReq	SINT	Set the required record number Value range 1 to 64, no sign.
Scale	INT	The controller uses a position factor and depending on the axis type this differs. For Linear Axis (DGCI, DNCI) the exponent of “10-2” is used. For example, if the Actual position of the axis is 123.67mm, the PLC Position data would be 12367. For the DSML, the exponent is “10-1” is used. For example, if the Actual position of the axis is 123.6°, the PLC Position data would be 1236. The embedded code inside the function block now uses the “Scale” instead of the hardcoded “100” for display of “actual value (position units)”

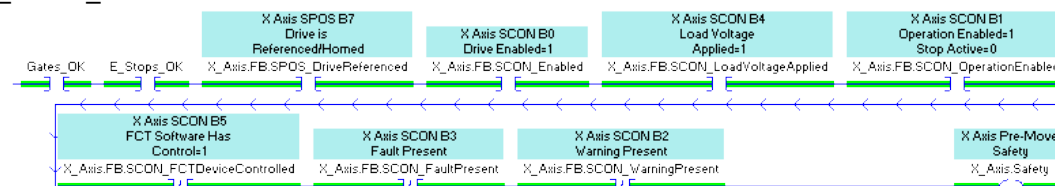
## CMAX\_FB\_MOVE\_RAM Description of Parameters for PLC Input

Input/Output	Type	Description
AcknowledgeStart	BOOL	= 0: Ready for start = 1: Start movement executed
AxisIsMoving	BOOL	Movement monitoring = 0: Drive is not moving (axis speed signal < limit), = 1: Drive is moving
FollowingError	BOOL	Following error or tolerance monitoring = 0: No following error / within tolerance = 1: Following error active / outside of tolerance
StandstillWarning	BOOL	= 0: Standstill warning Inactive = 1: Standstill warning active, drive moved after MC (position control)g not active
RecordChainActive	BOOL	If at least one step of record chaining has been configured: = 0: The first step criterion has not yet been reached = 1: The first switch or 2 <sup>nd</sup> record was executed/started.
RecordChainComplete	BOOL	= 0: Switching condition not met, record chaining aborted. = 1: Record chain was processed to the end of the chain.
Position_Force_State	BOOL	=0: Position Control Active, =1: Force Control Active
RecordChainError	BOOL	= 0: No error during record switching or no record switching programmed = 1: Record switching was programmed, but not executed. Record chaining aborted. A fault is signaled.
VelocityLimitReached	BOOL	= 0: Speed limit not reached, = 1: Speed limit reached
StrokeLimitReached	BOOL	= 0: Stroke limit not reached, = 1: Stroke limit reached, malfunction is signaled
FaultPresent	REAL	=0: No Fault, =1: Fault Present
MotionComplete	BOOL	= 0: Force job Active = 1: Target Force Achieved, (MC is set for the first time after switch-on [status "Drive disabled"])
RecordNumStatus	SINT	Last or Current record number executed Value range 1 to 64, no sign.
ActualValue	REAL	Primary Actual Position in actual units (NOTE: Please see previous page description on Parameter "Scale"). This is also determined by the setting of the "FeedbackTypePosition" parameter. (Actual value of the position in the set system of units depending on the operating status and the FHPP setting PNU 523)

# CMAx\_FB\_MOVE\_RAM Example Code

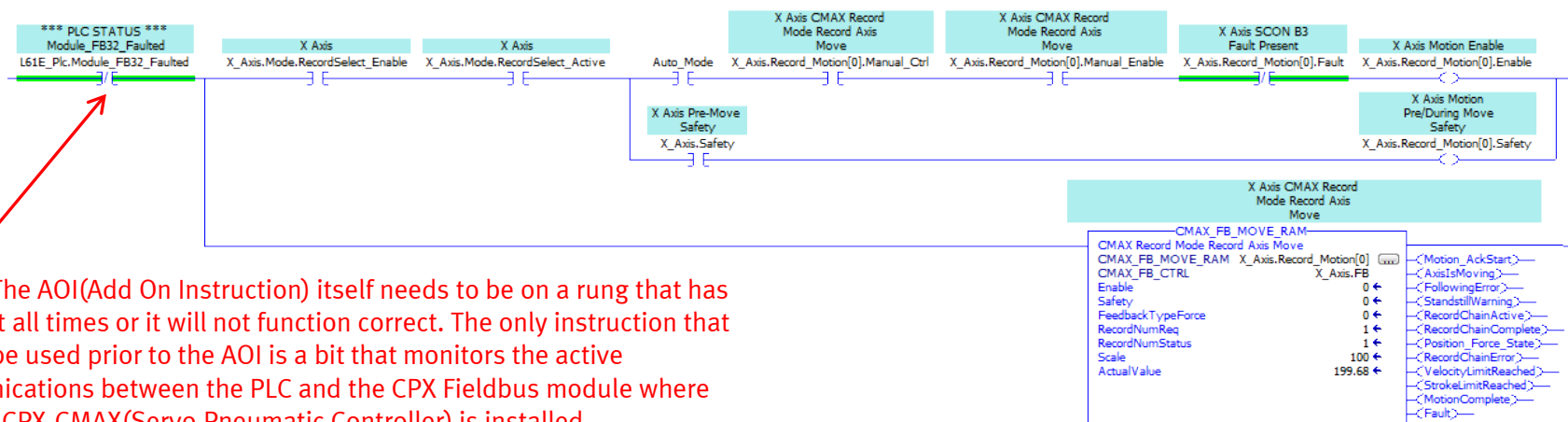
This instruction **executes one positional move at one time** and the user needs to assure that multiple calls do not happen simultaneously. The external PLC code structure operating on the CMAx\_FB\_MOVE\_PAM should be as shown below.

Before activation of the “Enable”, the “Safety” parameter needs to be true (prior to and during the move). Some common monitoring functions are shown in the safety rung here.



Prior to setting the “Enable”, the PLC needs to verify the following:

- ❑ communication to the CPX-CMAx is operational (L61E\_Plc.Module\_FB32\_Faulted)
- ❑ direct mode is active (Mode.RecordSelect\_Active).
- ❑ correct data is populated in such parameters as RecordNumReq.



NOTE: The AOI(Add On Instruction) itself needs to be on a rung that has power at all times or it will not function correct. The only instruction that should be used prior to the AOI is a bit that monitors the active communications between the PLC and the CPX Fieldbus module where the CPX-CPX-CMAx(Servo Pneumatic Controller) is installed.

## CMAX\_FB\_MOVE\_RAM Example Code

The below chart provides an example on how to logically perform moves in this mode of operation from the PLC/external higher order controller.

User action	Acknowledgement
PLC Selects <b>Record Mode</b> : CCON_OPM1_OperationMode = 0, CCON_OPM2_OperationMode = 0	<b>R_e_c_o_r_d_M_o_d_e_A_c_t_i_v_e=1</b> SCON_OPM1_OperationState = 0, SCON_OPM2_OperationState = 0 RecordSelect_Active= 1
PLC verifies Pre Enable functions	Manual_Ctrl = 1 or Manual_Ctrl = 0 depending on auto mode of desire to activate instruction manually by force Fault = 0 Safety = 1
PLC/User configures desired Options: Select feedback type: FeedbackTypeForce = 0 (Position Feedback)	No acknowledgement
RecordNumReq = 4 (Stored Record to execute within controller)	No acknowledgement
PLC requests Task Start: Enable= 0 → 1	Motion_AckStart = 1 (Motion is executing) AxisIsMoving = 1 (Drive is moving), FollowingError = 0 (no following error exists), RecordChainActive = 0 (Initially 0 until the first chain is encountered) RecordChainComplete = 0 (Off until all records within chain are completed) Position_Force_State = 1 (Force Mode Active), RecordChainError = 0 (Off due to lack of errors) VelocityLimitReached = 0 (Velocity Limit was NOT Reached) StrokeLimitReached = 0 (Stroke Limit was NOT Reached) MotionComplete = 0 , ActualValue = xxx (current position)
PLC resets Task Start request: Enable= 0	After completing positioning: Motion_AckStart = 0, AxisIsMoving = 0, RecordChainComplete = 1 (All records within chain are completed) MotionComplete = 1 StandstillWarning = 1 (If Axis moves after Motion complete)