

## Functional Description

Name: AMP Profinet drive setup with Siemens PLC and use of AMP Function Blocks library

Note: - This is not the complete list of the function blocks, some of the function blocks are still under development.

Note:- This guide assumes that the user has basic knowledge of TIA Portal.

























Software Used: TIA Portal V17

Communication Protocol: Profinet

Hardware Compatibility:

- Applied Motion Products drives with Profinet compatibility.
- Siemens PLC

Available Function Block list: -

▼  AMP_Drive	
 AMP_FC [FC2]	
 AMP_Drive_Alarms [FB2]	
 AMP_Drive_Crash_Stop [FB10]	
 AMP_Drive_Disable [FB5]	
 AMP_Drive_Enable [FB4]	
 AMP_Drive_Inputs [FB3]	
 AMP_Drive_Jog [FB9]	
 AMP_Drive_Reset [FB7]	
 AMP_Drive_SCL_Cmd [FB13]	
 AMP_Drive_Status [FB1]	
 AMP_Drive_Update_Speed [...]	

### References:

- [Host Command Reference Manual](#)
- Application Note APPN0061: "EthernetIP Function Blocks for CODESYS®"  
([Application Notes \(applied-motion.com\)](#))

# 1 Contents

Functional Description .....	1
<b>1 INTRODUCTION.....</b>	<b>4</b>
1.1 Additional Content .....	4
1.2 Reference Documentation .....	5
1.3 Example Hardware .....	6
<b>2 FUNCTION BLOCKS.....</b>	<b>7</b>
2.1 Applied Motion Profinet Function Block list.....	7
2.2 AMP_IO_Mapping Logic .....	8
2.2.1 Overview .....	8
2.2.2 Functionality .....	9
2.2.3 Parameters .....	9
2.3 AMP_Drive_Inputs Function Block.....	10
2.3.1 Overview .....	10
2.3.2 Functionality .....	11
2.3.3 Parameters .....	11
2.4 AMP_Drive Status Function Block .....	12
2.4.1 Overview .....	12
2.4.2 Functionality .....	12
2.4.3 Parameters .....	13
2.5 AMP_Drive_Alarms Function Block .....	14
2.5.1 Overview .....	14
2.5.2 Functionality .....	14
2.5.3 Parameters .....	15
2.6 AMP_Drive_Enable Function Block .....	16
2.6.1 Overview .....	16
2.6.2 Functionality .....	16
2.6.3 Parameters .....	16
2.7 AMP_Drive_Disable Function Block .....	18
2.7.1 Overview .....	18
2.7.2 Functionality .....	18




2.7.3	<i>Parameters</i> .....	18
2.8	AMP_Drive_Reset Function Block .....	19
2.8.1	<i>Overview</i> .....	19
2.8.2	<i>Functionality</i> .....	19
2.8.3	<i>Parameters:</i> .....	20
2.9	AMP_SCL_Command_Execute Function Block .....	21
2.9.1	<i>Overview</i> .....	21
2.9.2	<i>Functionality</i> .....	21
2.9.3	<i>Exceptions/Notes</i> .....	22
2.9.4	<i>Parameters:</i> .....	22
2.10	AMP_Drive_Move (Jog) Function Block .....	23
2.10.1	<i>Overview</i> .....	23
2.10.2	<i>Functionality</i> .....	23
2.10.3	<i>Exceptions/Notes</i> .....	24
2.10.4	<i>Parameters</i> .....	24
2.11	AMP_Update_Jog_Speed Function Block .....	25
2.11.1	<i>Overview</i> .....	25
2.11.2	<i>Functionality</i> .....	25
2.11.3	<i>Parameters</i> .....	26
2.12	AMP_Crash_Stop Function Block .....	27
2.12.1	<i>Overview</i> .....	27
2.12.2	<i>Functionality</i> .....	27
2.12.3	<i>Parameters:</i> .....	28

# 1 INTRODUCTION

This note will show how to use Applied Motion Products function blocks with Siemens PLC.

## 1.1 Additional Content

Within the zip file that contained this PDF, you can find the following other contents:

 AMP Siemens PLC Library Guide.pdf	Adobe Acrobat Document	1,804 KB	No	2,259 KB	21%
 AMP_lib_TIAV17.zal17	ZAL17 File	494 KB	No	521 KB	6%
 AMP_TIAV17_Sample_Project_0624.zap17	ZAP17 File	900 KB	No	904 KB	1%

1. Inside zip file AMP lib TIAV17.zal17 is library file to use in TIA Portal project with AMP function blocks.
2. A file named AMP TIAV17 Sample Project0624.zap17 which is an archive example program with all the Function Blocks imported, IO mapping and one pre-configured drive.

- This demo file assumes a drive IP addresses of 10.10.10.10
- This demo file assumes Siemens S7-1200 PLC IP addresses of 10.10.10.91
- SCALANCE Ethernet switch IP addresses of 10.10.10.93

## 1.2 Reference Documentation

This document only explains each Function Block's purpose and functionality. It is not intended to be fully inclusive of all the drive's features and functionality. For more information regarding drive commands, input/output data structures, acronyms, operands, etc. please refer to:

- i. [Host Command Reference Manual](#)
- ii. AMP YouTube video channel link: [Applied Motion Products, Inc. - YouTube](#)
- iii. Application Note APPN0061: "EthernetIP Function Blocks for Codesys®" ([Application Notes \(applied-motion.com\)](#)) // [EtherNet/IP Devices with CODESYS Tutorials](#).
- iv. Specific drive hardware manuals. These can be found in the Downloads section of the drive Product Page on the Applied Motion Products website ([Manuals\(applied-motion.com\)](#))

### 1.3 Example Hardware

This note was developed and primarily tested on the following hardware:

- Siemens PLC S7-1200 IP address 10.10.10.91
- Applied Motion Products SV2D10-IP – IP address 10.10.10.10
- Scalence Ethernet Switch – IP address 10.10.10.93

For determining compatibility of individual commands and features used by the Function Blocks with other hardware from Applied Motion Products, always refer to the Host Command Reference manual.

- Network Diagram



Figure 1: Network Diagram Example

## 2 FUNCTION BLOCKS

Following is a comprehensive list of all Function Blocks included in the ZIP folder accompanying this Note:

### 2.1 Applied Motion Profinet Function Block list.

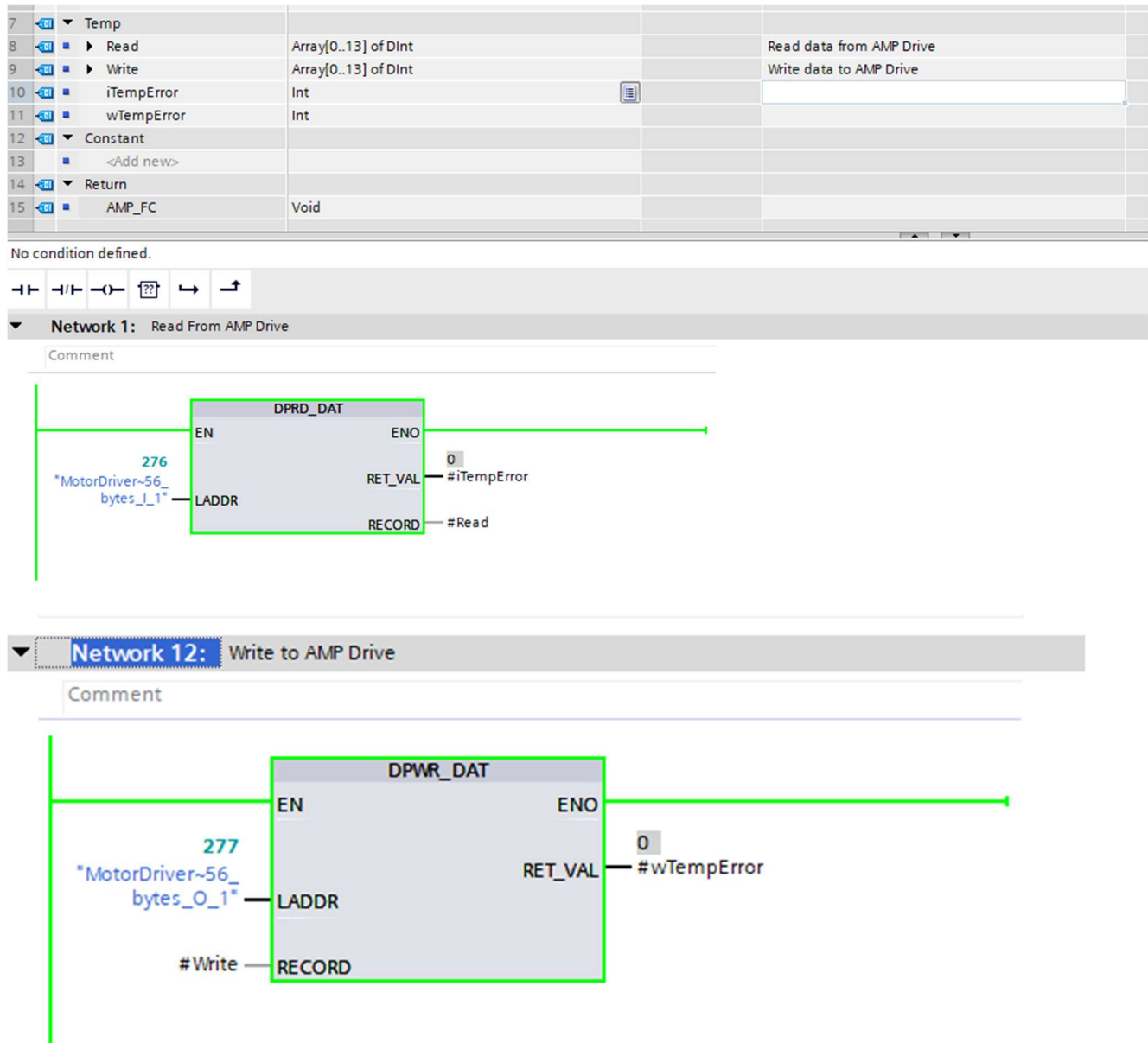
▼ AMP_Drive	●
AMP_FC [FC2]	●
AMP_Drive_Alarms [FB2]	●
AMP_Drive_Crash_Stop [FB10]	●
AMP_Drive_Disable [FB5]	●
AMP_Drive_Enable [FB4]	●
AMP_Drive_Inputs [FB3]	●
AMP_Drive_Jog [FB9]	●
AMP_Drive_Reset [FB7]	●
AMP_Drive_SCL_Cmd [FB13]	●
AMP_Drive_Status [FB1]	●
AMP_Drive_Update_Speed [...]	●

Figure 2: Available Function Blocks

## 2.2 AMP\_IO\_Mapping Logic

### 2.2.1 Overview

The AMP\_IO\_Mapping uses Siemens DPRD\_DAT and DPWR\_DAT function block to read and write the data to the drive. This logic is shown in AMP\_FC.





*Figure 3: AMP\_IO\_Mapping*

### 2.2.2 Functionality

- Define Read and Write as Temp inside FC. This will be use in function blocks as IO mapping.

### 2.2.3 Parameters

Name	Usage	Data Type	Description
Read	Temp	ARRAY [0..13] OF DINT	Read Values from Drive in DINT
Write	Temp	ARRAY [0..13] OF DINT	Write values to Drive in DINT

## 2.3 AMP\_Drive\_Inputs Function Block

### 2.3.1 Overview

This Function Block takes raw input data from the drive and re-interprets the data, using either scaling or copying, to appropriately named outputs.

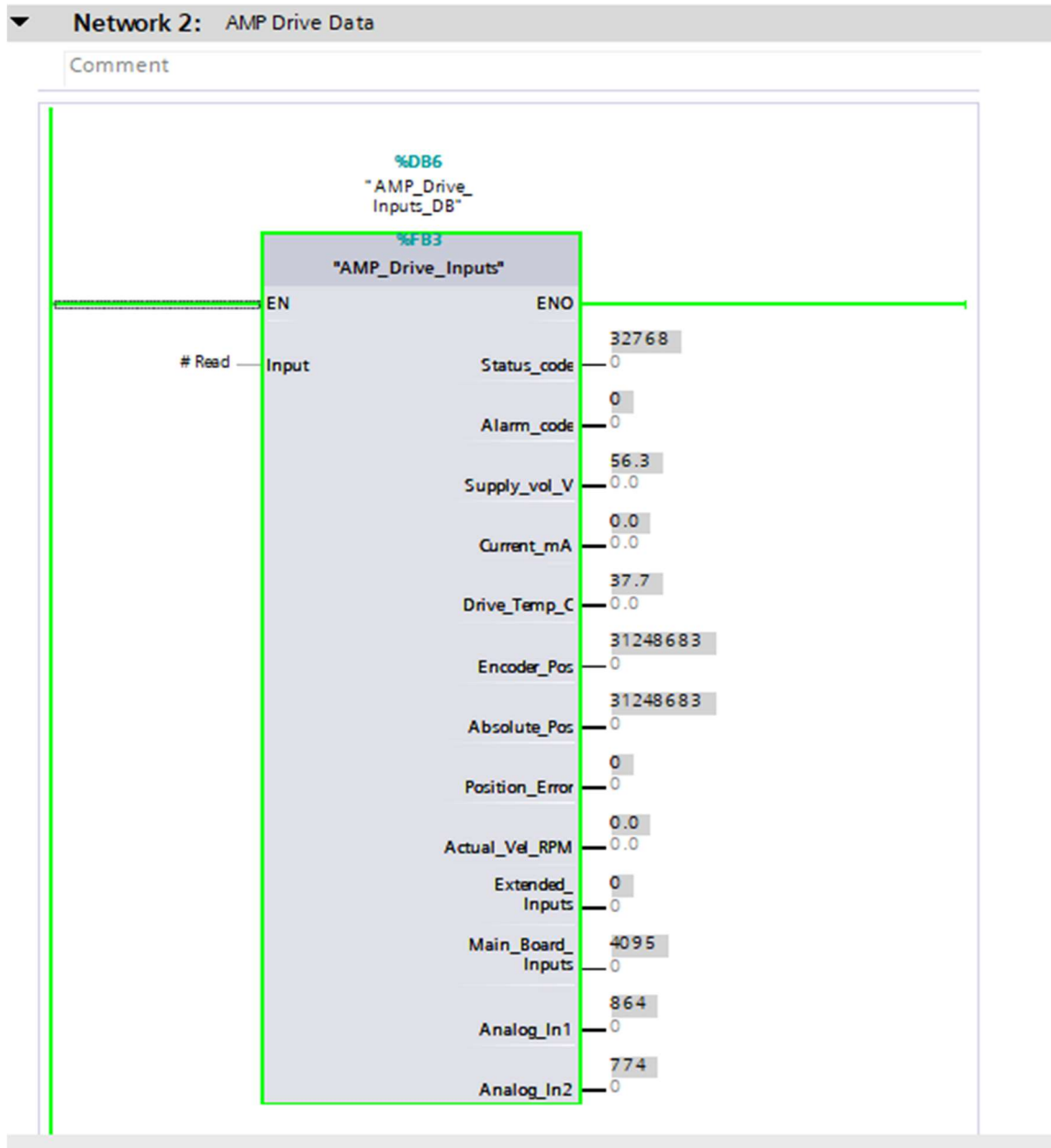


Figure 4: Function Block as Seen in Ladder Logic in TIA Portal

### 2.3.2 Functionality

The Input Assembly Function Block does not use the typical command sequence as it does not require sending any data to the drive. Here are the steps:

- The Function Block copies the raw data from the input assembly (Input parameter), in this example it is READ[0]( First DINT of READ)
- The Function Block copies and/or scales data from the Input and outputs it to named fields.

### 2.3.3 Parameters

Name	Usage	Data Type	Description
Input	InOut	ARRAY [0..13] OF DINT	Drive Input Assembly
Status_Code	Output	DINT	Status Code
Alarm_Code	Output	DINT	Alarm Code
Voltage	Output	REAL	Supply Voltage (V)
Current	Output	REAL	Actual Current (mA)
Temp	Output	REAL	Drive Temperature (1.0°C)
Encoder_Position	Output	DINT	Encoder Position (20,000 counts = 1 shaft turn)
Absolute_Position	Output	DINT	Absolute Position (20,000 counts = 1 shaft turn)
Velocity	Output	REAL	Actual Velocity (rev/sec, 0 when no encoder)
Position Error	Output	REAL	Position error from the encoder
Entended_Inputs	Output	INT	Input Status (Extended)
Main_Board_Inputs	Output	INT	Input Status (Main board)
Analog_Input_1	Output	DINT	Analog Input 1 (ADC counts, 0=min V, 16383 = max V)
Analog_Input_2	Output	DINT	Analog Input 2 (ADC counts, 0=min V, 16383 = max V)

## 2.4 AMP\_Drive Status Function Block

### 2.4.1 Overview

The Status Code Function Block uses the raw data stored in the Status Code word (Element 1 of Input Assembly. For further details, see page 242 of Ref. 1) to set individual appropriately named Boolean outputs. The equivalent drive command is SC.

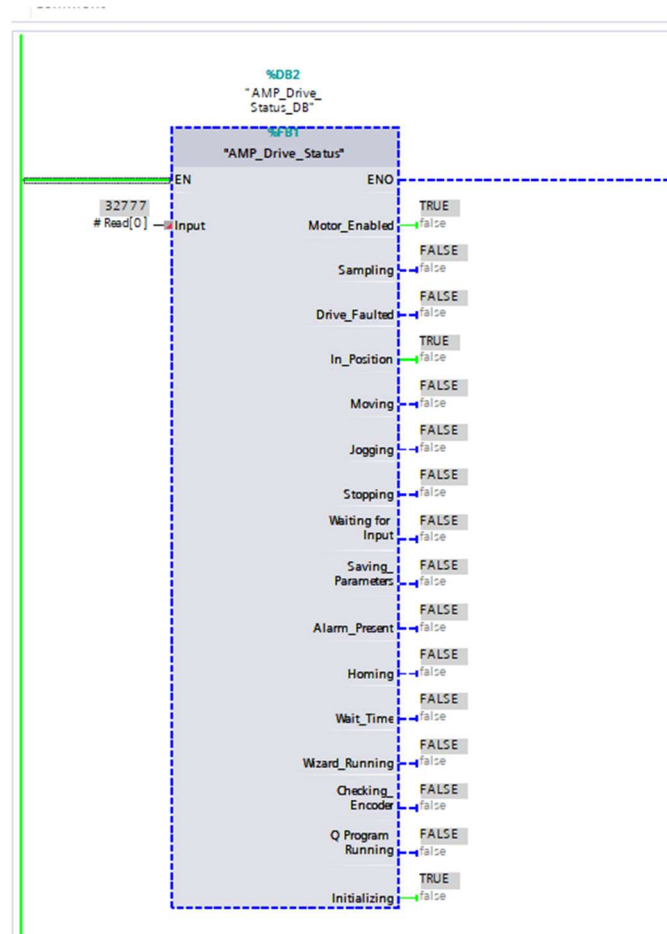


Figure 5: Applied Motion Status Code Function Block

### 2.4.2 Functionality

The Status Code Function Block does not use the typical command sequence as it does not require sending any data to the drive. Here are the steps:

- The Function Block copies individual Booleans from the Status Code word (Element 0 from Input Assembly 0x65) and sets appropriately named outputs.

### 2.4.3 Parameters

Name	Usage	Data Type	Description
Input	Input	ARRAY [0..13] OF DINT	Drive Status Code word (Element 0 from Input Assembly 0x65)
Enabled	Output	BOOL	Motor Enabled (Motor Disabled if this bit = 0)
Sampling	Output	BOOL	Sampling (for Quick Tuner)
Drive_Fault	Output	BOOL	Drive Fault (check Alarm Code)
In_Position	Output	BOOL	In Position (motor is in position)
Moving	Output	BOOL	Moving (motor is moving)
Jogging	Output	BOOL	Jogging (currently in jog mode)
Stopping	Output	BOOL	Stopping (in the process of stopping from a stop command)
Waiting	Output	BOOL	Waiting (for an input; executing a WI command)
Saving	Output	BOOL	Saving (parameter data is being saved)
Alarm_Present	Output	BOOL	Alarm present (check Alarm Code)
Homing	Output	BOOL	Homing (executing an SH command)
Wait_Time	Output	BOOL	Waiting (for time; executing a WD or WT command)
Wizard_Running	Output	BOOL	Wizard running (Timing Wizard is running)
Checking_Encoder	Output	BOOL	Checking encoder (Timing Wizard is running)
Q_Program_Running	Output	BOOL	Q Program is running
Initializing	Output	BOOL	Initializing (happens at power up) ; Servo Ready (for SV200 drives only)

## 2.5 AMP\_Drive\_Alarms Function Block

The Alarm Function Block uses the raw data stored in the Alarm Code word (Element 1 from Input Assembly) to set individual appropriately named Boolean outputs. The equivalent drive command is AC.

### 2.5.1 Overview

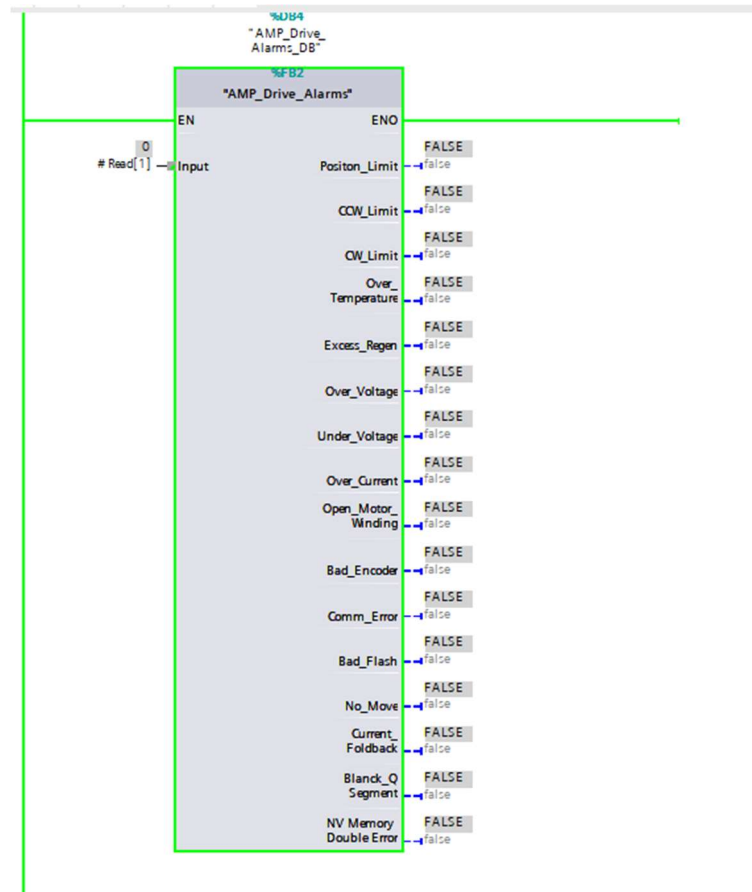


Figure 6: Applied Motion Alarm Code Function Block

### 2.5.2 Functionality

The Alarm Code Function Block does not use the typical command sequence as it does not require sending any data to the drive. Here are the steps:

- The Function Block copies individual Booleans from the Alarm Code word (Element 1 from Input Assembly, as seen in Reference Documents HCR) and sets appropriately named outputs.

### 2.5.3 Parameters

Name	Usage	Data Type	Description
Input	Input	ARRAY [0..13] OF DINT	Drive Alarm Code word (Element 1 from Input Assembly)
Position_Limit	Output	BOOL	Position Limit
CCW_Limit	Output	BOOL	CCW Limit
CW_Limit	Output	BOOL	CW Limit
Over_Temp	Output	BOOL	Over Temperature
Excess_Regen_or_Internal_Voltage	Output	BOOL	Excess Regen or Internal Voltage
Over_Voltage	Output	BOOL	Over Voltage
Under_Voltage	Output	BOOL	Under Voltage
Over_Current	Output	BOOL	Over Current
Open_Motor_Windings	Output	BOOL	Bad Hall Sensor / Open Motor Windings
Bad_Encoder	Output	BOOL	Bad Encoder
Comm_Error	Output	BOOL	Comm Error
Bad_Flash	Output	BOOL	Bad Flash
Wizard_Failed	Output	BOOL	Wizard Failed / No Move
Current_Foldback	Output	BOOL	Current Foldback / Motor Resistance
Blank_Q_Segment	Output	BOOL	Blank Q Segment
No_Move	Output	BOOL	No Move

## 2.6 AMP\_Drive\_Enable Function Block

### 2.6.1 Overview

The Drive Enable Function Block restores drive current to motor. The equivalent drive command is ME.

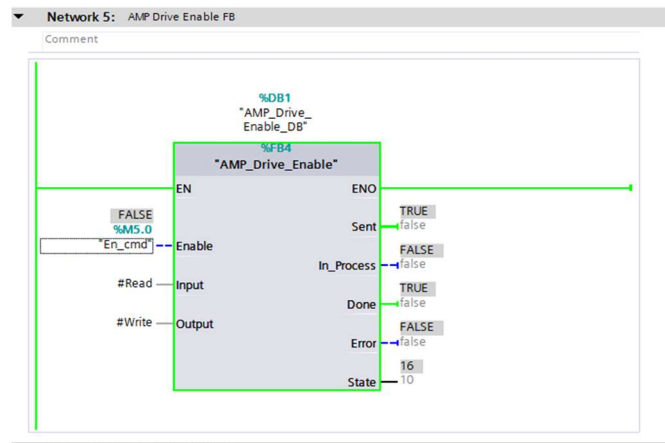


Figure 7: Applied Motion Motor Enable Function Block

### 2.6.2 Functionality

The Drive Enable Function Block uses a typical command sequence with the following specific differences:

- Parameters output to drive: none
- Drive Command: "Motor Enable" (command word value(write[0] in our example) of 0x2).
- Done output: when drive status indicates it is "Enabled".

Type: Buffered

### 2.6.3 Parameters

Name	Usage	Data Type	Description
Input	InOut	ARRAY [0..13] OF DINT	Drive Input Assembly
Output	InOut	ARRAY [0..13] OF DINT	Drive Output Assembly
Enable	Input	BOOL	Enables the drive(on rising edge)
Sent	Output	BOOL	Enable has been sent to drive
In_Progress	Output	BOOL	Enable in progress



## AMP Profinet drive function blocks guide for Siemens PLC

---

Done	Output	BOOL	Enable completed without errors – Motor is Enabled
Error	Output	BOOL	Enable failed to complete

## 2.7 AMP\_Drive\_Disable Function Block

### 2.7.1 Overview

The Motor Disable Function Block disables motor outputs (reduces motor current to zero). The equivalent drive command is MD.

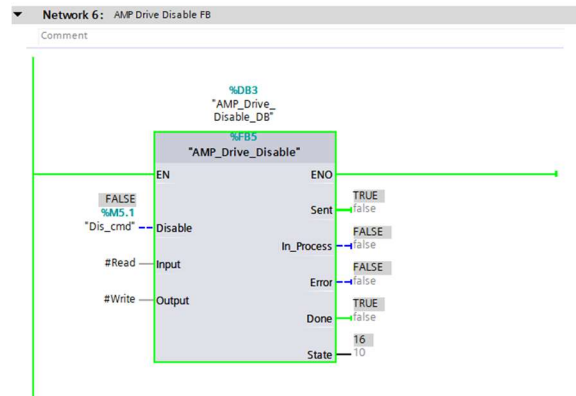


Figure 8: Applied Motion Motor Disable Function Block

### 2.7.2 Functionality

The Motor Disable Function Block uses a typical command sequence with the following specific differences:

- Parameters output to drive: none
- Drive Command: "Motor Disable" (command word value of 0x4).
- Done output: when drive status indicates it is not "Enabled".
- Type: Buffered

### 2.7.3 Parameters

Name	Usage	Data Type	Description
Input	InOut	ARRAY [0..13] OF DINT	Drive Input Assembly
Output	InOut	ARRAY [0..13] OF DINT	Drive Output Assembly
Disable	Input	BOOL	Disables the drive(on rising edge)
Sent	Output	BOOL	Disable has been sent to drive
In_Progress	Output	BOOL	Disable in progress

Done	Output	BOOL	Disable completed without errors – Motor is not Enabled
Error	Output	BOOL	Disable failed to complete

## 2.8 AMP\_Drive\_Reset Function Block

### 2.8.1 Overview

The Drive Reset Function Block is used to reset Drive Fault and clear Alarm Code (if possible). The equivalent drive command is AR.

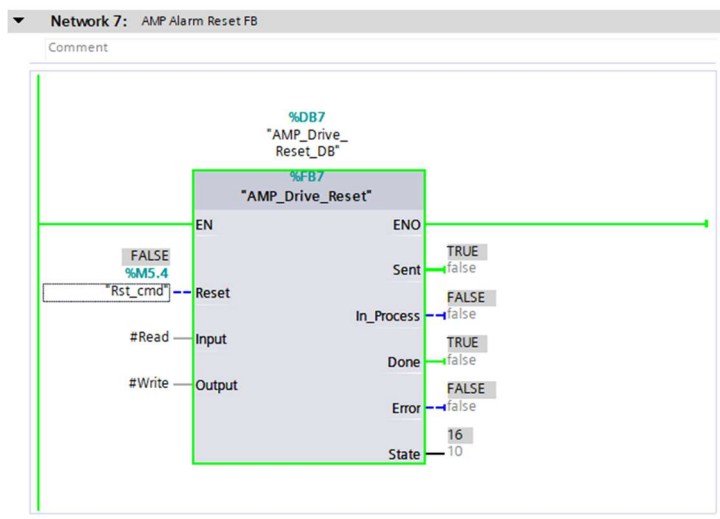


Figure 9: Applied Motion Alarm Reset Function Block

### 2.8.2 Functionality

The Alarm Reset Function Block uses a typical command sequence with the following specific differences:

- Parameters output to drive: not changed.
- Drive Command: "Alarm Reset" (AR) (command word value of 0x100000).
- Done output: when drive no longer indicates "Drive Faulted" or "Alarm Present".
- Type: Immediate.

### 2.8.3 Parameters:

Name	Usage	Data Type	Description
Input	InOut	ARRAY [0..13] OF DINT	Drive Input Assembly
Output	InOut	ARRAY [0..13] OF DINT	Drive Output Assembly
Reset	Input	BOOL	Reset the Drive (on rising edge)
Sent	Output	BOOL	Reset has been sent to drive
In_Progress	Output	BOOL	Reset in progress
Done	Output	BOOL	Reset completed without errors – no alarm present
Error	Output	BOOL	Reset failed to complete
State	Output	DINT	Internal state of progress

## 2.9 AMP\_SCL\_Command\_Execute Function Block

### 2.9.1 Overview

The SCL Command Execute Function Block executes selected SCL commands (as listed under “Available SCL Commands” in the Host Command Reference).

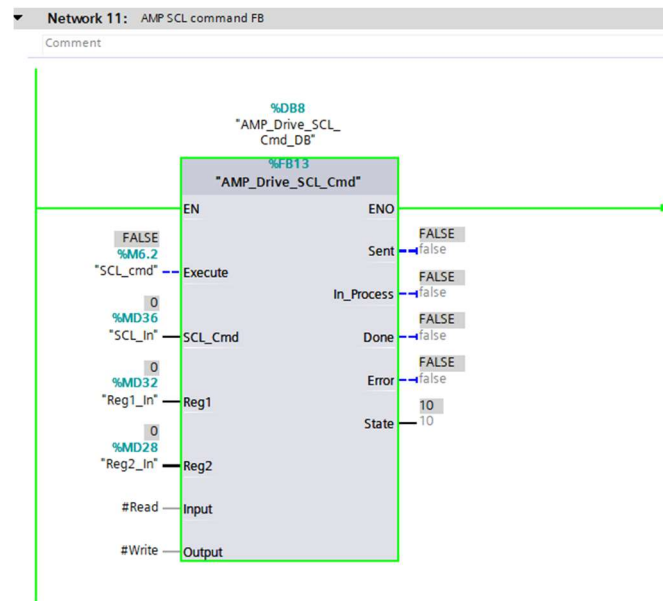


Figure 10: Applied Motion SCL Command Execute Function Block

### 2.9.2 Functionality

The SCL Command Execute Function Block uses a typical command sequence with the following specific differences:

- Parameters output to drive: SCL Command sent as two ASCII characters into lower 16 bits of output assembly element 9. First character is copied into bits 8-15, second character is copied into bits 0-7. SCL Data/Registers Reg1 and Reg2 parameters are copied unchanged to output assembly elements 10 and 11.
- Drive Command: the command word value is set to 0x40000.
- Done output: when command is sent.
- Type: Immediate

### 2.9.3 Exceptions/Notes

The user can set Reg1 and/or Reg2 parameters to ASCII values when required by enclosing the character value in single quotes.

### 2.9.4 Parameters:

Name	Usage	Data Type	Description
Input	InOut	ARRAY [0..13] OF DINT	Drive Input Assembly
Output	InOut	ARRAY [0..13] OF DINT	Drive Output Assembly
Execute	Input	BOOL	Execute the SCL command
SCL_Command	Input	STRING(2)	SCL Command Characters (always 2)
Reg1	Input	DINT	Data / SCL Register 1*
Reg2	Input	DINT	Data / SCL Register 2*
Sent	Output	BOOL	Command has been sent to drive
In_Progress	Output	BOOL	In progress = sending command
Done	Output	BOOL	Command sent
Error	Output	BOOL	Command failed to be sent
State	Output	DINT	Internal state of progress

\* Reg1 and/or Reg2 parameters can be set to ASCII values when required by enclosing the character value in single quotes.

## 2.10 AMP\_Drive\_Move (Jog) Function Block

### 2.10.1 Overview

The Drive Move Function Block is used to start jogging an axis at a specified speed and direction (a negative speed set point implies a CCW direction). The equivalent drive commands are JA, JL, JS and CJ.

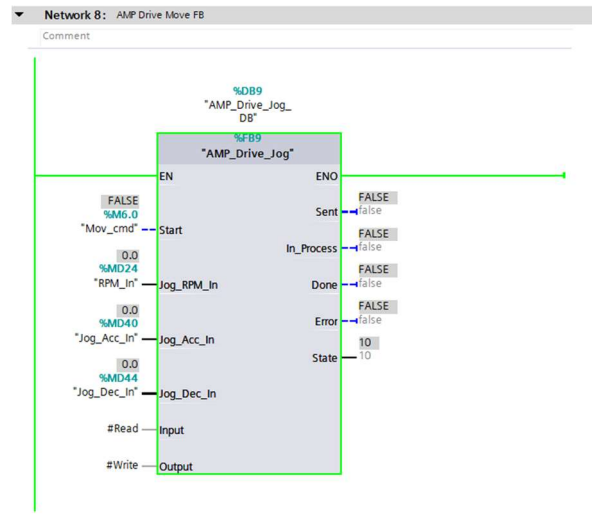


Figure 11: Applied Motion Jog Move Function Block

### 2.10.2 Functionality

The Jog Move Function Block uses a typical command sequence with the following specific differences:

- Parameters output to drive: The correctly scaled values of parameters Speed, Acc and Dec are written to the output assembly elements(write defined in the example) 1, 2 and 3, respectively.
- Drive Command: "Start Jogging" (command word value of 0x10000).
- Done output: when drive status indicates it is "Moving".
- Type: Buffered

### 2.10.3 Exceptions/Notes

The Jog Move only uses the rising edge of the Start parameter to start the jogging. To stop the motion, the user will need to use either the AMP\_Normal\_Stop or the AMP\_Crash\_Stop Function Block. Alternately, you may use AMP\_Update\_Jog\_Speed to set the speed to 0, this will indeed cause the motor to stop, but the drive will still report that it is jogging – this may cause confusion in some cases.

### 2.10.4 Parameters

Name	Usage	Data Type	Description
Input	InOut	ARRAY [0..13] OF DINT	Drive Input Assembly
Output	InOut	ARRAY [0..13] OF DINT	Drive Output Assembly
Start	Input	BOOL	Start Jogging*
Speed	Input	REAL	Velocity of Move (rev/s) (negative = CCW)
Acc	Input	REAL	Acceleration of Move (rev/s/s)
Dec	Input	REAL	Deceleration (rev/s/s)
Sent	Output	BOOL	Move has been sent to drive
In_Progress	Output	BOOL	Move in progress
Done	Output	BOOL	Move completed without errors - desired position reached
Error	Output	BOOL	Move failed to complete - drive not ready or faulted
State	Output	DINT	Internal state of progress

\*The Jog Move Function Block uses the rising edge of the Start parameter to start the jogging. To stop the motion, use either the AMP\_Normal\_Stop Function Block or the AMP\_Crash\_Stop Function Block.



## 2.11 AMP\_Update\_Jog\_Speed Function Block

### 2.11.1 Overview

The Update Jog Speed Function Block is used to modify the speed of an ongoing jog motion. The equivalent drive command is CS.

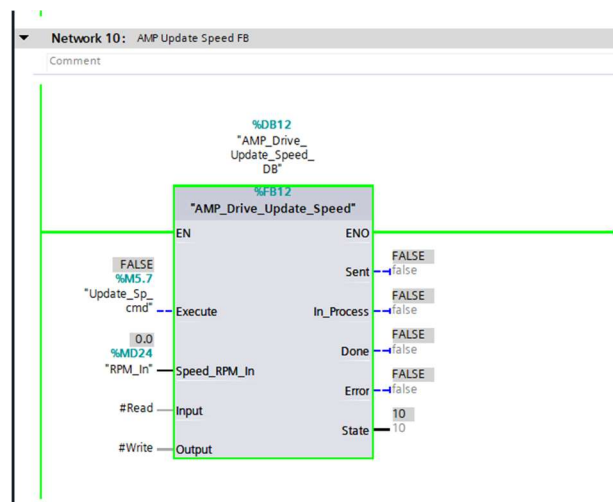


Figure 12: Applied Motion Update Jog Speed Function Block

### 2.11.2 Functionality

The Update Jog Speed Function Block uses a typical command sequence with the following specific differences:

- Parameters output to drive: The correctly scaled values of parameter “Speed” is written to the output assembly element 1.
- Drive Command: "Update Jog Speed" (command word value of 0x20000).
- Done output: immediately when command is sent.
- Type: Immediate

### 2.11.3 Parameters

Name	Usage	Data Type	Description
Input	InOut	ARRAY [0..13] OF DINT	Drive Input Assembly
Output	InOut	ARRAY [0..13] OF DINT	Drive Output Assembly
Update	Input	BOOL	Update the Jogging Speed
Speed	Input	REAL	Updated Jog Speed (rev/s), (negative = CCW)
Sent	Output	BOOL	Update has been sent to drive
In_Progress	Output	BOOL	Update in progress
Done	Output	BOOL	Update completed without errors
Error	Output	BOOL	Update failed to complete - drive not ready or faulted
State	Output	DINT	Internal state of progress

## 2.12 AMP\_Crash\_Stop Function Block

### 2.12.1 Overview

The Crash Stop Function Block is used to stop and kill the drive using the maximum deceleration as set by the AM command.

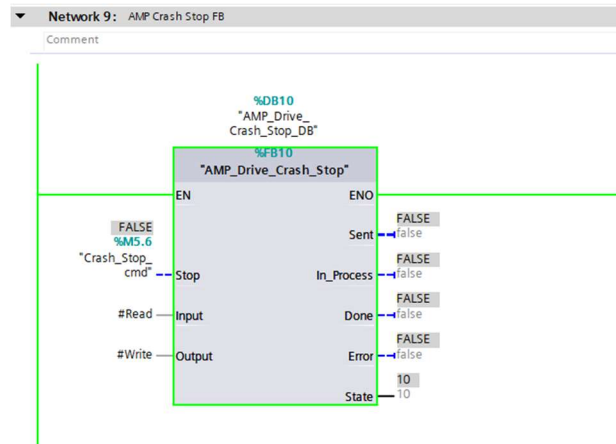


Figure 13: Applied Motion Crash Stop Function Block

### 2.12.2 Functionality

The Crash Stop Function Block uses a typical command sequence with the following specific differences:

- Parameters output to drive:- none.
- Drive Command: "Stop/Kill - AM" (command word value of 0x4000).
- Done output: when drive status indicates it is no longer "Moving".
- Type: Immediate.

### 2.12.3 Parameters:

Name	Usage	Data Type	Description
Input	InOut	ARRAY [0..13] OF DINT	Drive Input Assembly
Output	InOut	ARRAY [0..13] OF DINT	Drive Output Assembly
Stop	Input	BOOL	Stop the Drive (on rising edge)
Sent	Output	BOOL	Stop has been sent to drive
In_Progress	Output	BOOL	Stop in progress
Done	Output	BOOL	Stop completed without errors – Not moving
Error	Output	BOOL	Stop failed to complete (Always false – Future use only)
State	Output	DINT	Internal state of progress