



PROFINET SET UP GUIDE

Riccardo Piccinini
Mini Motor S.P.A.
FBS-EPN-EN
Firmware Revision: >4.052

Version 3, 04-2026

Table of Contents

- 1. ProfiNET 1
 - 1.2. Process Data 1
- 2. Application Example 7
 - 2.1. Tools 7
 - 2.2. Overview 7
 - 2.3. Basic Application 7
 - 2.4. Example Explanation 10

1. ProfiNET



Note: available only for drivers equipped with Profinet optional board (---/---/EPN).

This document outline the details of the Profinet implementation for Minimotor DBS55 integrated servomotor. By enabling Profinet/DSP402 "reference mode", the servodrive is commanded by means of CanOPEN DSP402 state machine and profiles.

For further information, please refer to: • DBS55 user manual: information about parametrization through USB and PC interface • Dsp402 v3: information about device state machine, control word/status word encoding and object dictionary.

1.1. Device identity

- Vendor_ID: 0x0419
- Device_ID: 0x0001

1.2. Process Data

There are four possible configuration of the Ethernet IP process data.

- **Default(DFL)**: the standard process data implementing the profile velocity/torque/position/homing
- **A**: Standard process data with the possibility to read Digital and Analog inputs and command via fieldbus the Digital Out. Parameter P015 must be on 10-Fieldbus.
- **B**: A process data with the Cia 402 Touch probe functionality
- **C**: A process data with our manufacturer custom Touch Probe implementation
- **D**: A process data with rms acceleration values of the three axis

You must select the corresponding device description and set the P176 parameter to the desired value.

1.2.1. Layout Default (DFL)

The configuration for process data is the following:

PLC Outputs (42 Bytes)

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x00	U16	ControlWord	0x6040	0000h	see DSP402
0x02	S32	TargetPosition	0x607A	+0	position units
0x06	U16	TorqueLimit	0x6072	1000	× 0.1% Trq
0x08	S32	TargetVelocity	0x60FF	+0	rpm
0x0C	S16	TargetTorque	0x6071	+0	× 0.1% Trq
0x0E	U8	ModeOfOperation	0x6060	1	Profile ^[1]
0x0F	U8	Padding Byte			
0x10	U32	ProfileVelocity	0x6081	+3000	rpm
0x14	U32	ProfileAccel	0x6083	+3000	rpm/s
0x18	U32	ProfileDecel	0x6084	+3000	rpm/s
0x1C	S32	PositionLimit.Neg	0x607D.1	-2147483648	position units
0x20	S32	PositionLimit.Pos	0x607D.2	+2147483647	position units
0x24	S32	HomingOffset	0x607C	+0	position units
0x28	S8	HomingMethod	0x6098	37	SetQuota ^[1]
0x29	U8	Padding Byte			

PLC Inputs (18 Bytes)

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x00	U16	StatusWord	0x6041	0250h	Motor Status ^[1]
0x02	S32	PositionActualValue	0x6064	+0	position units
0x06	S16	TorqueActualValue	0x6077	+0	× 0.1% Trq
0x08	S32	VelocityActualValue	0x606C	+0	rpm
0x0C	U16	VDclink	0x3010	+0	× 0.1 V
0x0E	U8	ModeOfOperation	0x6060	1	Profile ^[1]
0x0F	U8	Padding Byte			
0x10	U16	ActualAlarm	0x3008	0	see DBS Manual

1.2.2. Layout A

The configuration for process data is the following:

PLC outputs (44 Bytes)

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x00	U16	ControlWord	0x6040	0000h	see DSP402
0x02	S32	TargetPosition	0x607A	+0	position units
0x06	U16	TorqueLimit	0x6072	1000	× 0.1% Trq
0x08	S32	TargetVelocity	0x60FF	+0	rpm
0x0C	S16	TargetTorque	0x6071	+0	× 0.1% Trq
0x0E	U8	ModeOfOperation	0x6060	1	Profile ^[1]
0x0F	U8	Padding Byte			
0x10	U32	ProfileVelocity	0x6081	+3000	rpm
0x14	U32	ProfileAccel	0x6083	+3000	rpm/s
0x18	U32	ProfileDecel	0x6084	+3000	rpm/s

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x1C	S32	PositionLimit.Neg	0x607D.1	-2147483648	position units
0x20	S32	PositionLimit.Pos	0x607D.2	+2147483647	position units
0x24	S32	HomingOffset	0x607C	+0	position units
0x28	S8	HomingMethod	0x6098	37	SetQuota ^[1]
0x29	U8	Padding Byte			
0x2A	U16	Digital output	0x3002	0	bit0: DOUT1 bit1..15: not used ^[2]

PLC inputs (22 bytes)

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x00	U16	StatusWord	0x6041	0250h	Motor Status ^[1]
0x02	S32	PositionActualValue	0x6064	+0	position units
0x06	S16	TorqueActualValue	0x6077	+0	× 0.1% Trq
0x08	S32	VelocityActualValue	0x606C	+0	rpm
0x0C	U16	VDclink	0x3010	+0	× 0.1 V
0x0E	U8	ModeOfOperation	0x6060	1	Profile ^[1]
0x0F	U8	Padding Byte			
0x10	U16	ActualAlarm	0x3008	0	see DBS Manual
0x12	S16	Analog Input	0x3000	0	-32768 ... +32767
0x14	U16	Digital Input	0x3001	0	Bit0: DIN1 Bit1: DIN2 Bit2: DIN3 Bit3: DIN4 Bit4: DIN5 Bit5..15: not used

1.2.3. Layout B

The configuration for process data is the following:

PLC outputs (46 bytes)

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x00	U16	ControlWord	0x6040	0000h	see DSP402
0x02	S32	TargetPosition	0x607A	+0	position units
0x06	U16	TorqueLimit	0x6072	1000	× 0.1% Trq
0x08	S32	TargetVelocity	0x60FF	+0	rpm
0x0C	S16	TargetTorque	0x6071	+0	× 0.1% Trq
0x0E	U8	ModeOfOperation	0x6060	1	Profile ^[1]
0x0F	U8	Padding Byte			
0x10	U32	ProfileVelocity	0x6081	+3000	rpm
0x14	U32	ProfileAccel	0x6083	+3000	rpm/s
0x18	U32	ProfileDecel	0x6084	+3000	rpm/s
0x1C	S32	PositionLimit.Neg	0x607D.1	-2147483648	position units
0x20	S32	PositionLimit.Pos	0x607D.2	+2147483647	position units
0x24	S32	HomingOffset	0x607C	+0	position units
0x28	S8	HomingMethod	0x6098	37	SetQuota ^[1]

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x29	U8	Padding Byte			
0x2A	U16	Digital output	0x3002	0	bit0: DOUT1 bit1..15: not used ^[2]
0x2C	U16	Touch Probe Function	0x60B8	0	^[1]

PLC inputs (32 bytes)

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x00	U16	StatusWord	0x6041	0250h	Motor Status ^[1]
0x02	S32	PositionActualValue	0x6064	+0	position units
0x06	S16	TorqueActualValue	0x6077	+0	× 0.1% Trq
0x08	S32	VelocityActualValue	0x606C	+0	rpm
0x0C	U16	VDclink	0x3010	+0	× 0.1 V
0x0E	U8	ModeOfOperation	0x6060	1	Profile ^[1]
0x0F	U8	Padding Byte			
0x10	U16	ActualAlarm	0x3008	0	see DBS Manual
0x12	S16	Analog Input	0x3000	0	-32768 ... +32767
0x14	U16	Digital Input	0x3001	0	Bit0: DIN1 Bit1: DIN2 Bit2: DIN3 Bit3: DIN4 Bit4: DIN5 Bit5..15: not used
0x16	U16	Touch Probe Status	0x60B9	0	^[1]
0x18	S32	Touch Probe Rising Edge position	0x60BA	0	position units
0x1C	S32	Touch Probe Falling Edge position	0x60BB	0	position units

1.2.4. Layout C

The configuration for process data is the following:

PLC outputs (52 Bytes)

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x00	U16	ControlWord	0x6040	0000h	see DSP402
0x02	S32	TargetPosition	0x607A	+0	position units
0x06	U16	TorqueLimit	0x6072	1000	0.1% Trq
0x08	S32	TargetVelocity	0x60FF	+0	rpm
0x0C	S16	TargetTorque	0x6071	+0	0.1% Trq
0x0E	U8	ModeOfOperation	0x6060	1	Profile ^[1]
0x0F	U8	Padding Byte			
0x10	U32	ProfileVelocity	0x6081	+3000	rpm
0x14	U32	ProfileAccel	0x6083	+3000	rpm/s
0x18	U32	ProfileDecel	0x6084	+3000	rpm/s
0x1C	S32	PositionLimit.Neg	0x607D.1	-2147483648	position units
0x20	S32	PositionLimit.Pos	0x607D.2	+2147483647	position units
0x24	S32	HomingOffset	0x607C	+0	position units
0x28	S8	HomingMethod	0x6098	37	SetQuota ^[1]

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x29	U8	Padding Byte			
0x2A	U16	Digital output	0x3002	0	bit0: DOUT1 bit1..15: not used ^[2]
0x2C	S32	Touch Probe positioning offset	0x3020	0	Position Units
0x2C	S32	Touch Probe positioning modulo	0x3021	0	Position Units

PLC inputs (22 bytes)

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x00	U16	StatusWord	0x6041	0250h	Motor Status ^[1]
0x02	S32	PositionActualValue	0x6064	+0	position units
0x06	S16	TorqueActualValue	0x6077	+0	× 0.1% Trq
0x08	S32	VelocityActualValue	0x606C	+0	rpm
0x0C	U16	VDclink	0x3010	+0	× 0.1 V
0x0E	U8	ModeOfOperation	0x6060	1	Profile ^[1]
0x0F	U8	Padding Byte			
0x10	U16	ActualAlarm	0x3008	0	see DBS Manual
0x12	S16	Analog Input	0x3000	0	-32768 ... +32767
0x14	U16	Digital Input	0x3001	0	Bit0: DIN1 Bit1: DIN2 Bit2: DIN3 Bit3: DIN4 Bit4: DIN5 Bit5..15: not used

1.2.5. Layout D

The configuration for process data is the following:

PLC outputs (44 Bytes)

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x00	U16	ControlWord	0x6040	0000h	see DSP402
0x02	S32	TargetPosition	0x607A	+0	position units
0x06	U16	TorqueLimit	0x6072	1000	× 0.1% Trq
0x08	S32	TargetVelocity	0x60FF	+0	rpm
0x0C	S16	TargetTorque	0x6071	+0	× 0.1% Trq
0x0E	U8	ModeOfOperation	0x6060	1	Profile ^[1]
0x0F	U8	Padding Byte			
0x10	U32	ProfileVelocity	0x6081	+3000	rpm
0x14	U32	ProfileAccel	0x6083	+3000	rpm/s
0x18	U32	ProfileDecel	0x6084	+3000	rpm/s
0x1C	S32	PositionLimit.Neg	0x607D.1	-2147483648	position units
0x20	S32	PositionLimit.Pos	0x607D.2	+2147483647	position units
0x24	S32	HomingOffset	0x607C	+0	position units
0x28	S8	HomingMethod	0x6098	37	SetQuota ^[1]
0x29	U8	Padding Byte			

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x2A	U16	Digital output	0x3002	0	bit0: DOUT1 bit1..15: not used ^[2]

PLC inputs (28 Bytes)

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x00	U16	StatusWord	0x6041	0250h	Motor Status ^[1]
0x02	S32	PositionActualValue	0x6064	+0	position units
0x06	S16	TorqueActualValue	0x6077	+0	× 0.1% Trq
0x08	S32	VelocityActualValue	0x606C	+0	rpm
0x0C	U16	VDclink	0x3010	+0	× 0.1 V
0x0E	U8	ModeOfOperation	0x6060	1	Profile ^[1]
0x0F	U8	Padding Byte			
0x10	U16	ActualAlarm	0x3008	0	see DBS Manual
0x12	S16	Analog Input	0x3000	0	-32768 ... +32767
0x14	U16	Digital Input	0x3001	0	Bit0: DIN1 Bit1: DIN2 Bit2: DIN3 Bit3: DIN4 Bit4: DIN5 Bit5..15: not used
0x16	U16	Acceleration rms x	0x3031		
0x19	U16	Acceleration rms y	0x3032		
0x1A	U16	Acceleration rms z	0x3033		

^[1] see DSP402

^[2] P015=10 required for fieldbus operation

2. Application Example

2.1. Tools

Tia V20 Example

Legacy Tia 16 Example (not explained here)

BSI Parametrization software

2.2. Overview

This document show the creation of a DBS/FC smart motor application using Profinet communication interface; a Siemens 1215C CPU controller is configured and an application is written into the Tia Portal V16 development environment.

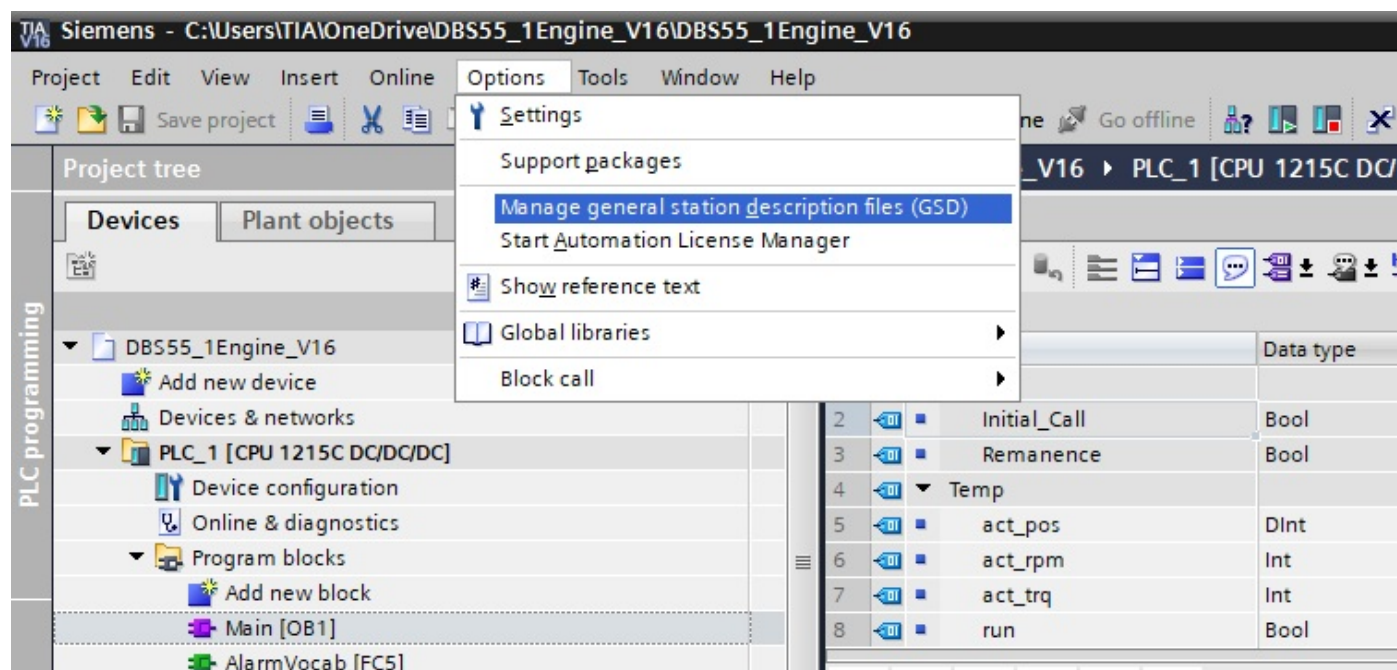
References:

- Siemens 1215C DC/DC/DC CPU
- DBS/FC Firmware version 4.041
- Tia Portal V19

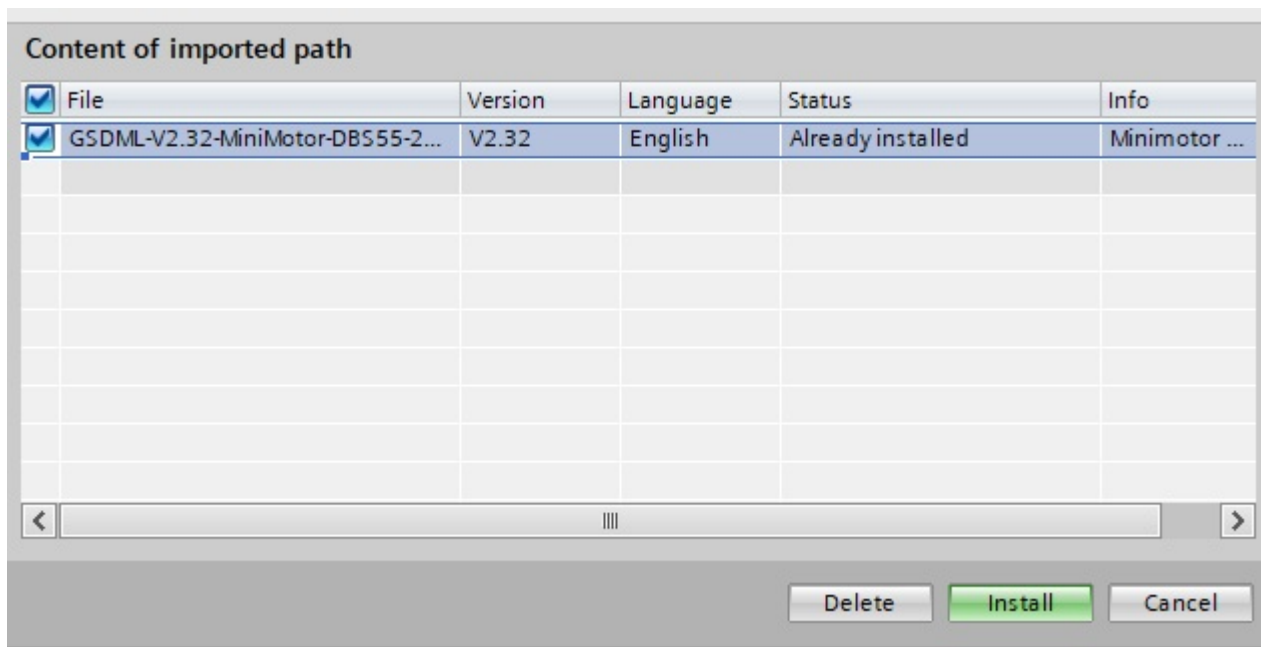
2.3. Basic Application

2.3.1. Install GSD

- Go to Options→Manage general station description files (GSD)

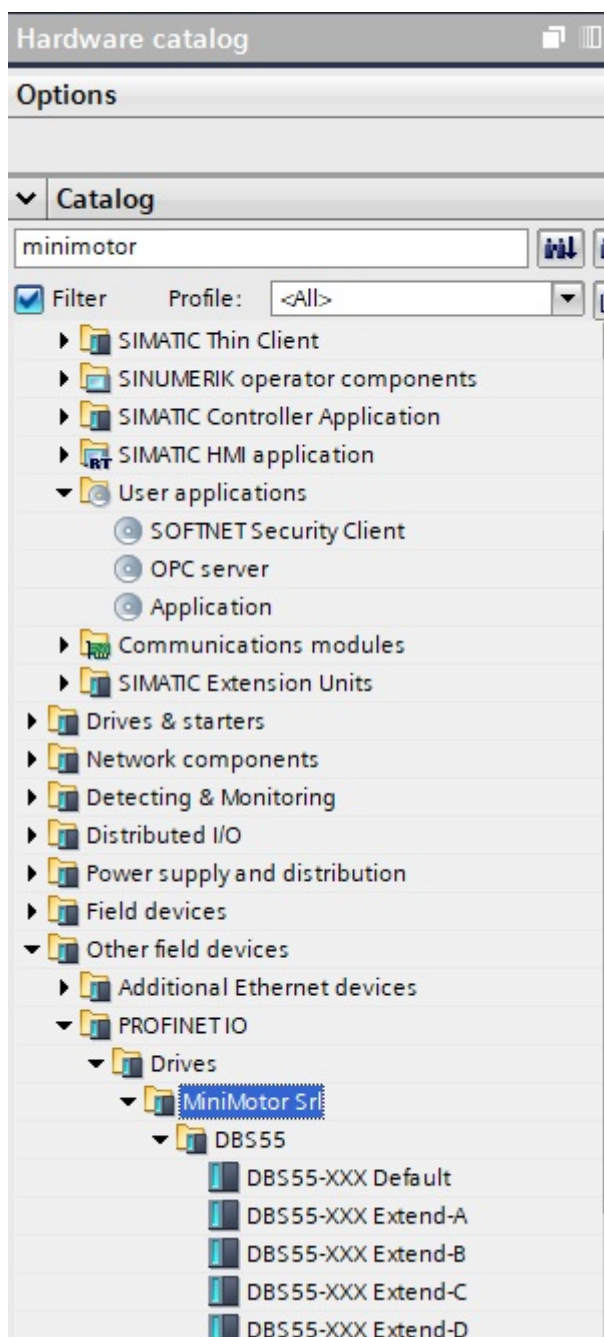


- Find the folder MM-DBS_vX.XXX\DeviceDesc\Profinet from the BSI Interface download
- Select the shown GSD and click on Install

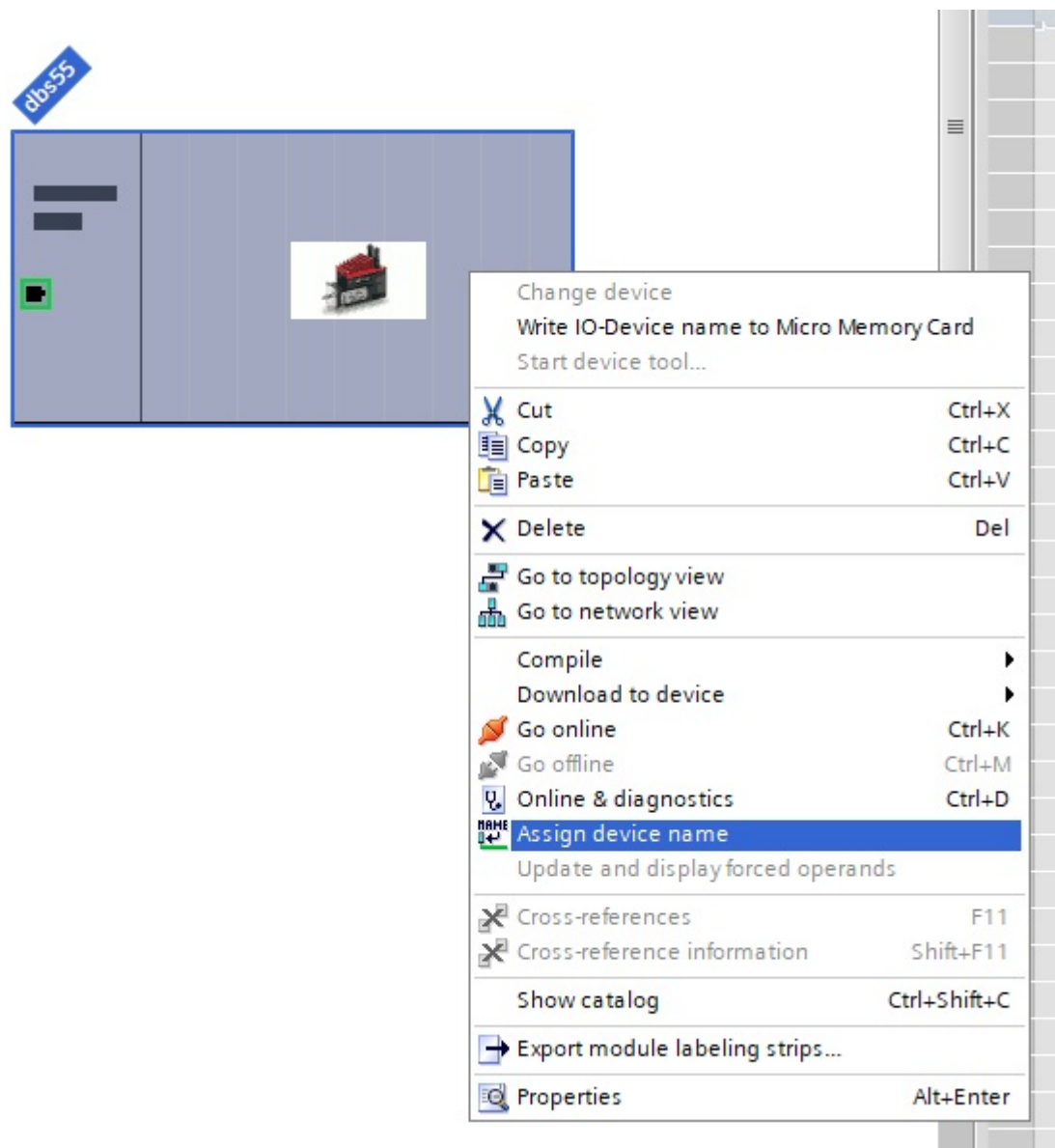


2.3.2. Add Mini Motor system to the project

- After having imported the GSD find in the Hardware catalog the motor



- Import in your project the DBS55-XXX file with the Process data you want to use. Refer to the first section of this guide for how these process data are formed. For the example, the Default Process Data is used.
- Once imported, remember to assign the device a name.



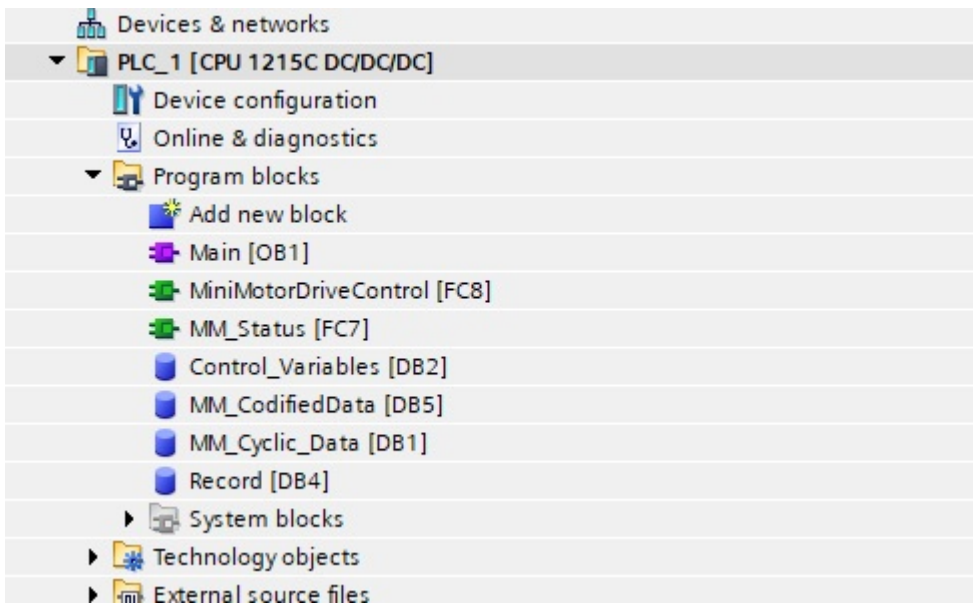
2.4. Example Explanation

Download and open the [Tia V19 Example](#)

We will take a look at how it works. It can be commanded via the variables in the MAIN code.

2.4.1. Mini Motor Elements

The Example is composed of 2 Mini Motor made FB and 2 Mini Motor made DB and 2 data types.



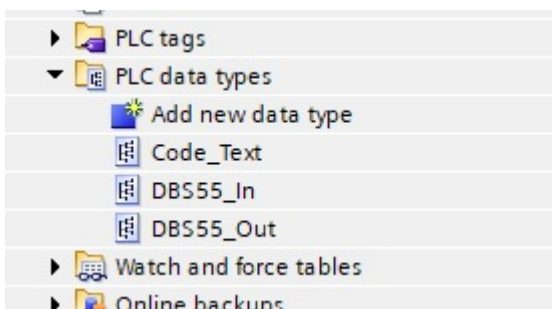
The MiniMotorDriveControl FC is responsible for all motor control. The MM_Status is a block to show readily available information at all times from the specific unit.

The MM_CodifiedData DB has strings and codes used in the Status block to give a text feedback on the motor status based on the input data.

The MM_Syclic_Data has a copy of the cyclic data coming from the motor area of memory, for local manipulation.

The other two DBs, Control_Variables and Record, are used and/or created automatically to support Siemens blocks or to support the example commands to be changed in real time.

The Data types are just the Motor Cyclic data packaged in a neat format.

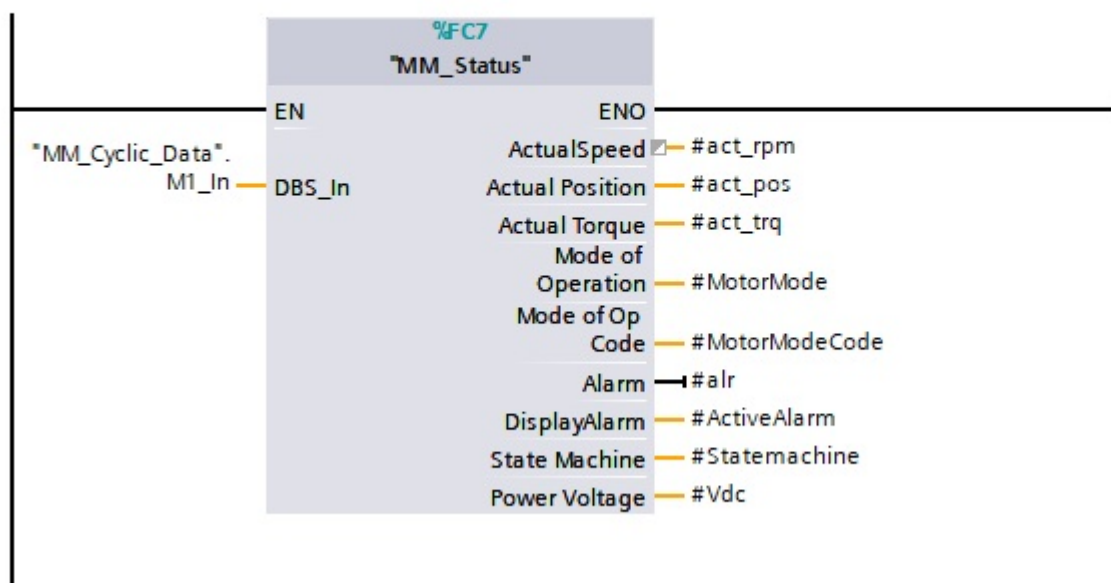


2.4.2. FC Explanations

MM_Status

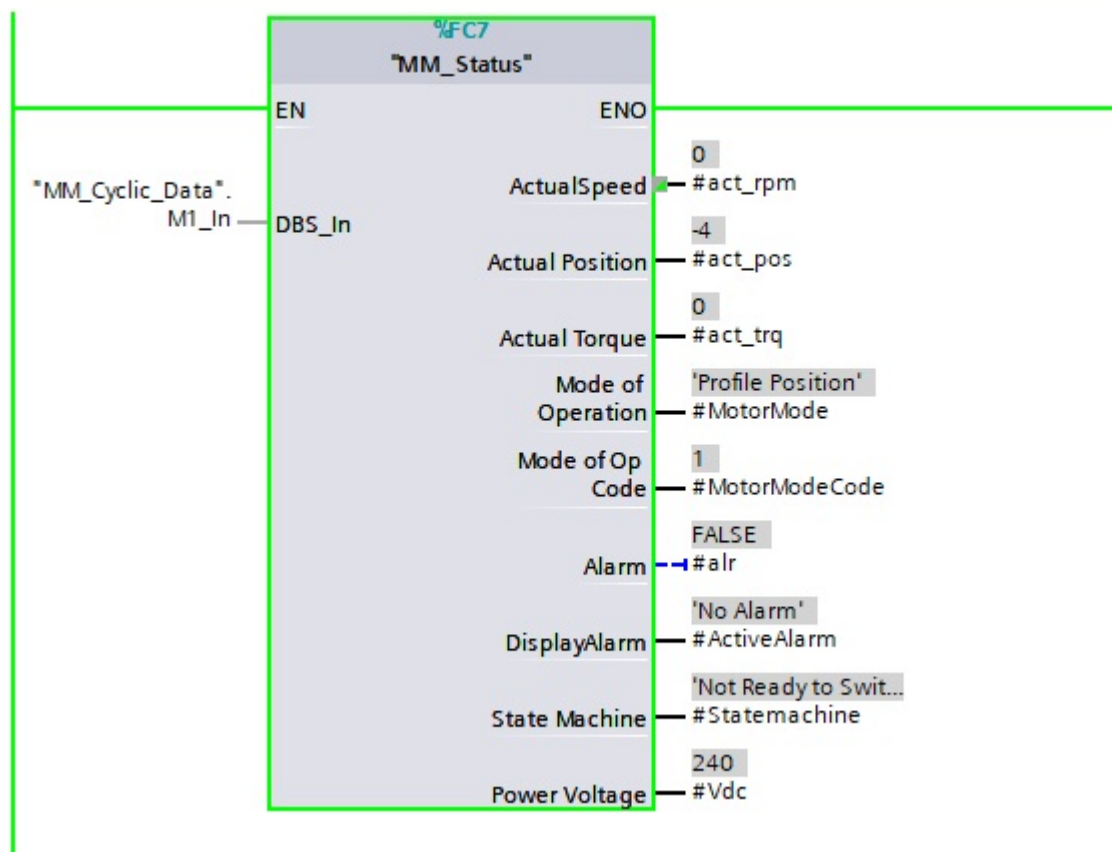
Network 2: Motor Status

- ▼ This block reads and interprets the data from the motors. Showing the actual values for speed, torque, position, but also which motor mode the motor is operating, which alarm it has and the state machine logic. Also the Power Voltage



The MM_Status FC takes as an input the MM_Cyclic_Data.M1_in, which is a local copy of the Motor image sent to the PLC, copied in the first network of the Main program.

The Code inside looks at the Status Word, voltage, mode of operation and other data useful for a feedback from the motor.



The outputs are:

- Actual Speed: The speed of the motor in rpm

- Actual Position: The position of the motor in encoder pulses, eventually scaled by the user using the motor internal parameters
- Actual Torque: the actual torque used by the motor. In thousands per rated nominal torque, so 1000 = 100% nominal.
- Mode of Operation: The name of the Operating mode the motor is currently in
- Mode of Op Code: The code of the current mode of operation
- Alarm: boolean value if there is or not an error on the drive
- Display Alarm: Text explaining the error
- State Machine: Where the motor is in the DS402 State machine. This is done by reading the Status word bits and the output is in text
- Power Voltage: The voltage reading of the Motor Power Pins.

With this you have a quick look at all the motor variables, that can be used for all kinds of program behaviour.

MiniMotorDriveControl

Network 3: The general Mini Motor Drive Control logic

- ▼ The Motor can work in Profile Position, Velocity, Torque and Homing. This is chosen by the Operating mode, following the DS402 numbering code for the other operating modes. The...



The MiniMotorDriveControl takes the inputs from the program and navigates the CanOpenDS402 commands automatically across all operating modes of Profile Velocity, Position, Torque and Homing to

send the correct commands to the motor.

The block needs the Cyclic Data In and Out to read and write to the motor. The Cyclic data out is then sent out to the motor memory area by the last network in the example.

The Motor has 3 main Inputs.

The input *EN* is used to power the FC and initialize/shutdown the motor. This way the motor will be kept in Ready to Switch On State.

The input *Enable Operation* will put the motor in Enable Operation, making it execute whichever command is given depending on the *modeofop* variable.

- 1: Profile Position
- 3: Profile Velocity
- 4: Profile Torque
- 6: Profile Homing

The input *Start Positioning/Home* is used to start the actual homing or positioning of the motors.

For further information, we demand the in depth explanation to the Canopen DS402 specification.

The other inputs set the variable used by the various modes.

Set Speed, *Set Torque*, *Acceleration*, *Deceleration* all contribute to the trajectory, with the velocity, maximum torque that the motor can apply and the acceleration and deceleration, for a trapezoidal profile.

Position Limit Neg and *Position Limit Pos* are used to limit via software the range of movement for the motor. When put at their maximum value they are disabled. These are -2^{31} and $2^{31}-1$

Target Position is used to set the target position to be reached by positioning mode when the *Start Positioning/Home* input sees a rising edge.

Change Immediately and *ABS/REL* are two options for profile positioning. The first one if true it will enable immediate change of target position while the motor is already doing a move, otherwise it will put the new move command in a buffer. The second if true enables relative positioning where the target pos is added to the current position. False is absolute movements.

The *Halt* command PAUSES the current job and when it's removed the motor will RESUME the previous command.

Home Method defines which homing method the motor will follow (see main manual), and *home offset* is the offset to add to the 0 position.

The outputs of this block are bits that can be used to monitor specific mode of operation behaviour. They are all boolean values self explanatory in their name. Be sure to correctly interpret them depending on the Mode of Operation the motor is in.

The block also enables the Alarm reset with the command *Fault Reset*

Async Read and Write

All the CanOpen (6000h) and Motor Manufacturer (2000h) parameters can be read and written with the RDREC and WRREC by Siemens. For more information about the parameter size, check the main manual.

To save the Manufacturer Parameters (2000h), it can be written the value 65766173h to the index 1010h, like shown in the picture. This command saves all changes, so it can be done last.

Parameters 6000h CANNOT be saved, so they must be loaded up at every startup.

