

Application note

Accessing drive parameters via EtherCAT SDO

AN00242

Rev C (EN)

Modify ABB EtherCAT servo drive parameters via EtherCAT at runtime via Service Data Object (SDO) access



Introduction

AC500 PLCs (PM585 and PM59x) can be used to perform real-time motion control of ABBs EtherCAT enabled servo drives. In some applications it is necessary to dynamically modify drive settings (e.g. control loop gains, sentinel configuration parameters, touchprobe/latch filter distances). Instead of consuming Process Data Object (PDO) mappings to achieve this, which would unnecessarily consume available EtherCAT cycle time, these settings/parameters can be modified by the AC500 PLC using Service Data Object (SDO) access instead.

This application note details how to use the AC500 programming environment within Automation Builder to write and read drive parameters via SDO.

Pre-requisites

You will need to have the following to work through this application note:

- Mint Workbench build 5860 or later (see new.abb.com/motion for latest downloads and support information)
- A MicroFlex e190 or MotiFlex e180 drive with build 5868 or later firmware
- A PC or laptop running Automation Builder 2.1.1 or later
- An installed copy of the ABB PLCopen motion control library (PS552-MC-E v3.2.0 or later)
- One of the following AC500 PLC processors.....PM585, PM590, PM591, PM592 or PM595 (PLC processors should be running firmware version 2.5.1 or later). The PM595 is provided with an integrated EtherCAT coupler (this should be running firmware version 4.2.32.2 or later). All other processors require a CM579-ECAT communication module (which must be running firmware version 2.6.9 or later, but ideally version 4.3.0.2 or later). Contact your local ABB PLC support team for details on how to check these requirements and update if necessary or visit <http://new.abb.com/plc/programmable-logic-controllers-plcs> and select the link for 'Software'. For the purposes of the text in this application note we have assumed the use of a PM591 PLC with CM579-ETHCAT coupler
- Ethernet cable to connect the EtherCAT coupler to the drive
- A copy of application note AN00205 (AC500 and ABB motion drives - EtherCAT Getting Started Guide) and the Automation Builder PLC project that is included with it

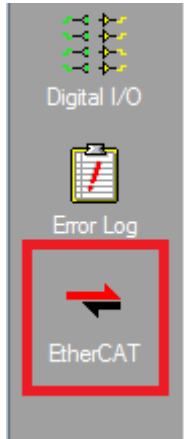
To follow the basic steps to create example code to perform SDO access only requires a PC or laptop running Automation Builder 2.1.1 or later and an installed copy of the PS552-MC-E motion control libraries. It is assumed the reader has a basic working knowledge of Mint Workbench, Automation Builder, CoDeSys and the AC500 PLC and that the reader has read and understood the contents of application note AN00205, which is also available for download from new.abb.com/motion, and has commissioned an EtherCAT based servo drive (MicroFlex e190 or MotiFlex e180 for example) ready for use with the AC500 PLC.

Enabling access to drive parameters

PLC library EtherCAT_AC500_V13.lib is automatically included in the Library Manager for the project when an EtherCAT coupler is included in the PLC’s hardware configuration. This library includes a function block called ECAT_COE_WRITE which can be used to perform a SDO write to any of the EtherCAT objects on the connected drive(s) without needing to include a PDO mapping. Please refer to the Automation Builder Help system for further information about this function block if required.

The drive objects that are available are listed in the Mint Workbench Help system (refer to the topics titled ‘EtherCAT protocol objects: 1000h – 2000h’ and ‘EtherCAT & Ethernet POWERLINK: e190 and e180 manufacturer specific objects’ for further details).

However, a more convenient way to view (and search for) these objects is to use the ‘EtherCAT’ page within Mint Workbench...

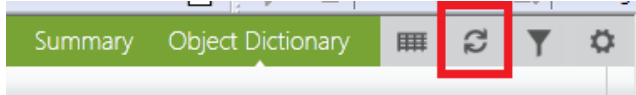


This page can be accessed during an online connection to a MicroFlex e190 or MotiFlex e180 drive, but can also be accessed by starting a Mint Workbench project in an ‘offline’ mode and selecting one of these two drives for offline operation.

Once you’ve clicked this icon Workbench will display some basic information about EtherCAT operation on the drive. At the top right of this EtherCAT window is a label named ‘Object Dictionary’.....click on this to display the Object Dictionary for the selected drive...



The available objects are listed in a (hexadecimal) numerical order. If you are connected to a drive you can view the current values by enabling refresh of the Object Dictionary view – this is done by clicking the ‘Refresh’ icon (showing two circular arrows) in the top right corner...



....and selecting an appropriate refresh rate (1s – for 1 second) is usually adequate...

Address	Name	Actual
Index: 1000 - NMT_DeviceType_U32 (1 item)		
1000:00	NMT_DeviceType_U32	131474 (16#0020192)
Index: 1001 - ERR_ErrorRegister_U8 (1 item)		
1001:00	ERR_ErrorRegister_U8	0 (16#00)
Index: 1008 - NMT_ManufactDevName_VS (1 item)		
1008:00	NMT_ManufactDevName_VS	MicroFlex e150
Index: 1009 - NMT_ManufactHwVers_VS (1 item)		
1009:00	NMT_ManufactHwVers_VS	E152A03E10A
Index: 100A - NMT_ManufactSwVers_VS (1 item)		
100A:00	NMT_ManufactSwVers_VS	MicroFlex e150 Build 5852.0.1 (WIP)
Index: 1010 - NMT_StoreParam_REC (2 items)		
1010:00	NMT_StoreParam_REC.SubIndex 000	1 (16#01)
1010:01	NMT_StoreParam_REC.AllParam_U32	1 (16#00000001)

Refresh panel details:

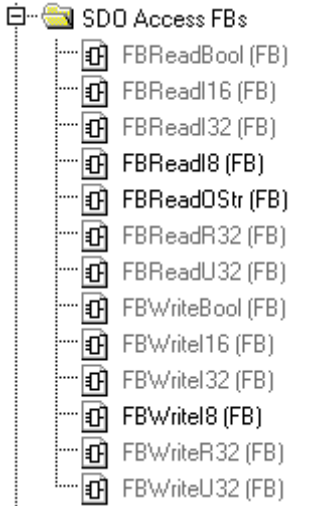
- Refresh button
- Load actual values: Update
- Load actual every: 1s
- Last update: 13:51:28

When the PLC writes to the drive’s EtherCAT objects this only sets the value on the “EtherCAT side” of the drive (i.e. in the object dictionary). These values do not automatically affect the “Mint side” parameters (i.e. those that can be seen using the ‘Parameter’ screen in Mint Workbench). To ensure that values written to the Object Dictionary are transferred across to the Mint parameters it is necessary for the PLC to write the value ‘TRUE’ to object 3004h (hex) subindex 00 (CFG_ImmediateApply_BOOL). Notice that the last part of the object name identifies the data type, it is important to recognize this when writing the PLC code necessary to access the drive objects (more on this later).

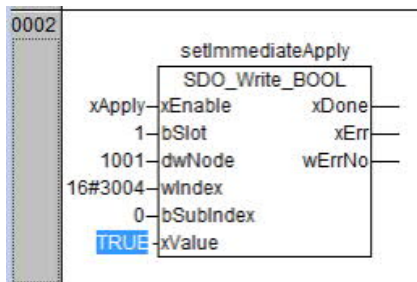
Once object 3004h has been set TRUE any data that is written to the EtherCAT objects is immediately applied to the associated Mint parameter (e.g. writing a value to object 405Bh subindex 01 with the immediate apply object set to TRUE would result in the value of SENTINELTRIGGERVALUEFLOAT(0,_stvLOW) being changed).

In most cases runtime changes like this are best kept as “volatile” changes (i.e. if the drive is power cycled the object value and the Mint parameter value will return to their original settings). If however it is necessary to write to an object and save this change permanently then it is necessary for the PLC to write the string ‘evas’ (‘save’ backwards) to object 1010h (hex) subindex 01 (NMT_StoreParam_REC.AllParam_U32). This object will automatically revert to a value of 1 when the save is complete.

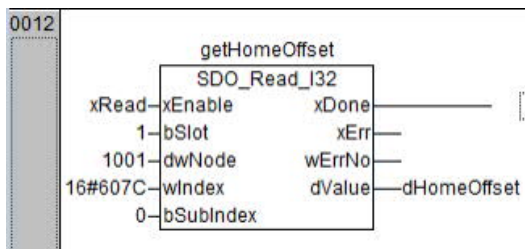
The PLC byte order for data differs from that used on the drive and it is therefore necessary to swap the order of the data being sent to the drive to match that used by the drive. To simplify this process an Export file containing function blocks to read and write drive data via SDO is included with this application note. To import these function blocks into your PLC program use Project>Import... from the CoDeSys menu and select the ‘SDO Access.exp’ file included with this document. The function blocks will be added to your PLC project in a new folder named ‘SDO Access FBs’ as shown below...



Example usage of one of the SDO write function blocks is shown below (in this case illustrating how the CFG_ImmediateApply_BOOL object may be set TRUE)....



The following screenshot shows example usage of one of the SDO read function blocks (in this case illustrating how to read the current home offset value on the drive)...



The operation of the input and output parameters is described as follows:

- xEnable – Rising edge on this input triggers the read or write
bSlot – Slot number of the EtherCAT coupler
dwNode – Node number of the EtherCAT slave/drive
wIndex – Object index to be read or written (use 16# to express this as a hexadecimal value)
bSubIndex – Subindex to the object index (as defined by the drive’s object dictionary)
xDone – Read/Write function block has completed (use this in conjunction with the xErr output to determine if it completed successfully or not). xDone is reset when xEnable resets
xErr – Indicates if an error occurred or not (TRUE/FALSE)

wErrNo – If an error occurs this will indicate an error code (refer to the Automation Builder Help system for further details on the function block error codes)

xValue/bValue/iValue/dValue/dwValue/rValue – this is the input (write) or output (read) value. Use a data type relevant to the specific function block type

A timeout of 2000ms (i.e. 2 seconds) is embedded into each function block, this may be modified by editing the function block source if required, but this value is set by default to match the drives' ESI file data (if attempting SDO access to the drive whilst Workbench is also accessing the drive object dictionary you may find the PLC SDO function blocks can take 300-400ms to complete so it is recommended to keep the 2 second timeout if possible).

Contact Us

For more information please contact your local ABB representative or one of the following:

new.abb.com/motion
new.abb.com/drives
new.abb.com/drivespartners
new.abb.com/PLC

© Copyright 2016 ABB. All rights reserved.
Specifications subject to change without notice.

EtherCAT® is a registered trademark and patented technology, licenced by Beckhoff Automation GmbH, Germany