Motion Control Products

Application note
Initialising an EtherCAT network

AN00243

Rev C (EN)

Function blocks provided as part of the PS552-MC-E motion control library can be used to detect the presence of all the configured EtherCAT hardware

Introduction

AC500 PLCs (PM585 and PM59x) can be used to perform real-time motion control of ABBs EtherCAT enabled servo drives and control distributed EtherCAT I/O devices. This application note details how to use the AC500 programming environment within Automation Builder to detect when the configured EtherCAT devices are operational on the network, ready to be accessed, and how to therefore interlock processing of any EtherCAT related code until these devices are ready.

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 (and licensed) 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 check the status of the EtherCAT devices 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.

Detecting configured devices

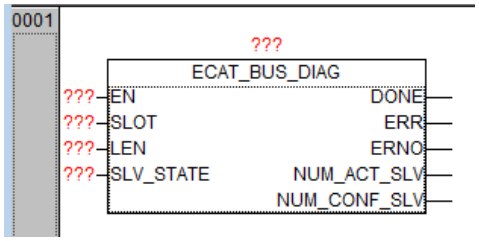
PLC library EtherCAT_AC500_V13.lib is automatically included in the Library Manager for the project when an EtherCAT coupler is included in the PLC's hardware configuration. This library includes a function block called ECAT_BUS_DIAG which can be used to read the status of, and to return status information for, each slave on the EtherCAT network. This function block can also detect the number of active slave devices connected to the EtherCAT coupler as well as reporting the number of configured devices expected from the Automation Builder hardware tree.

By comparing the number of active devices against the number of expected/configured devices the user's application program can detect when all EtherCAT devices are operational.

For this application note we will add our ECAT_BUS_DIAG function block to the project included with application note AN00205, but the same process can be applied to any AC500 motion program using an EtherCAT coupler/network.

Add a new network to a program unit called from the lower priority motion related task (in our case we added a network to the very start of the 'Motion' program in the PLC project from AN00205). It is logical to place this diagnostic code at the very start of the program as we can use the detection of all of the configured slaves to allow the rest of the code in this program section to execute (or not).

Add a new box to this network and enter ECAT_BUS_DIAG as the name of this function block (or use the Input Assistant/F2 to find this function block from the list of standard function blocks)…



The table below details the input and output parameters used with this function block…

Input	Description	Data type
EN	Enable – this input operates more like an 'Execute' input in that the function block outputs (e.g. NUM_ACT_SLV) are updated following a rising edge on the EN input. As it takes some time for the EtherCAT network to become operational it is therefore typical to use a BLINK function block to continually toggle the EN input TRUE/FALSE and examine the function block outputs on every transition of the DONE output	BOOL
SLOT	Slot number of the EtherCAT coupler	BYTE
LEN	Tells the function block how large the buffer in SLV_STATE is. SLV_STATE is a pointer to an array of bytes, where a byte is required for every slave device on the EtherCAT network	WORD
SLV_STATE	Pointer to an array of bytes. Each byte is used to store information about a connected EtherCAT slave device. For example, if there were 4 MicroFlex e190 drives connected to the EtherCAT coupler than LEN would be set to 4 and SLV_STATE would point to an array of 4 bytes	POINTER TO ARRAY OF BYTE
Output	**Description**	**Data type**
DONE	Indicates the processing state of the block. After completion or abortion of processing (due to an error) the DONE output is set TRUE for one program cycle. This output should always be considered in conjunction with the ERR output	BOOL
ERR	Indicates whether an error occurred during processing of the block	BOOL
ERNO	Provides an error identifier if ERR is set TRUE. See the Automation Builder Help system for further information about possible ERNO codes	WORD
NUM_ACT_SLV	Indicates the number of EtherCAT slaves active on the EtherCAT network (i.e. Operational and exchanging cyclic data with the AC500's EtherCAT coupler)	DWORD
NUM_CONF_SLV	Indicates the number of EtherCAT slaves configured via the Automation Builder devices tree (maximum number of slave devices is 256)	DWORD

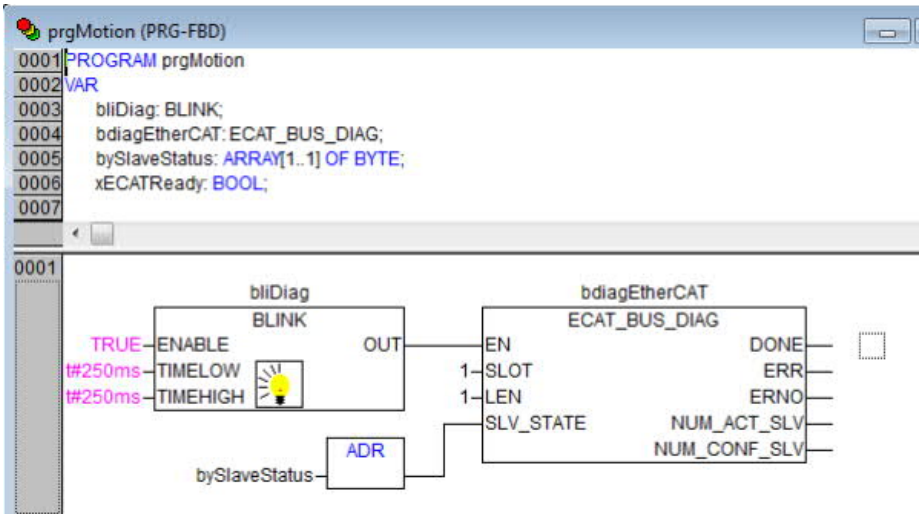
Each byte in the array of bytes pointed to by SLV_STATE indicates the actual state of the EtherCAT slave controller on a drive. Possible values for the byte are:

1 Init State
2 Pre-Operational State
3 Request Bootstrap State
4 Safe-Operational State
8 Operational State

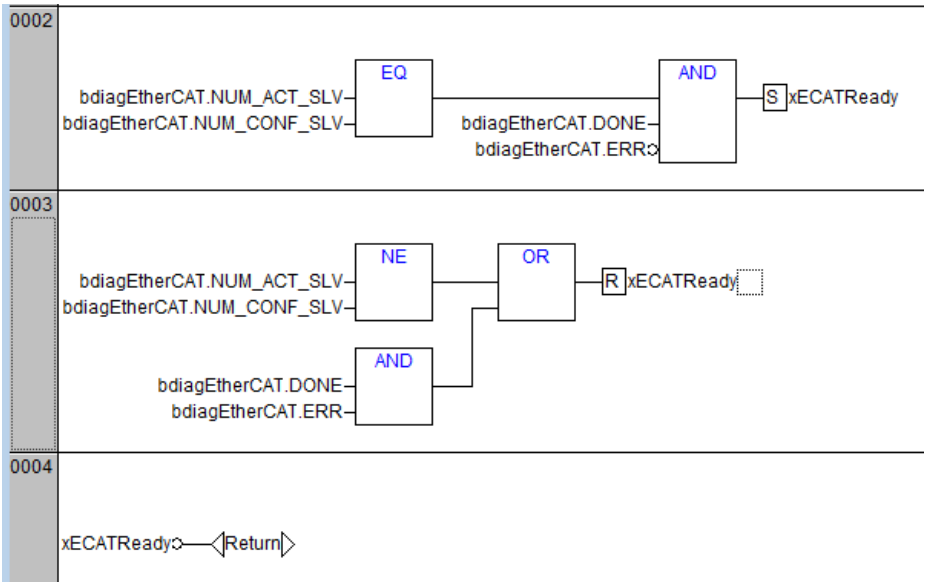
If a byte indicates a value other than 8 then it is certain that the slave device is not operational. However, just because the byte may indicate a value of 8 (Operational) it doesn't mean the slave device really is operational….the network could have an issue and the PLC may be unable to read the current status of the slave.

We therefore aren't typically too worried about examining the content of these bytes and instead just concentrate on the active nodes diagnostic.

For our example we only have a single drive connected to our EtherCAT coupler so our completed network and our associated variable declarations look like this…



We can then create some logic to set and reset an "EtherCAT ready" Boolean variable that we can use as an interlock in our motion program (e.g. we could exit the motion program before processing any of our motion function blocks if we've detected the network is not yet fully operational). The screenshot below illustrates some typical logic to fulfill this sort of functionality…



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

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