**Kollmorgen S700**

**Low Level EtherCAT Support**

1. **Introduction**

This document describes how the low level support of the Kollmorgen S700 EtherCAT drive works.

Due to non-compliance with the CiA-402 (DSP-402) specification only low level support will be provided for this drive.

1. **How to Setup**
   1. **EtherCAT Setup**

The first part of the setup is configuring the EtherCAT network. A modified S700 ESI XML file should have been provided along with this document that is configured as required for the low level support algorithm.

*Note: The low level support algorithm requires certain data objects to be read from / written to the S700 in real-time (PDOs). In order to cover all required functionality the fixed RxPDO 0x1704 was used, and the fixed TxPDO 0x1B08 was used.*

After placing the ESI XML file in the EtherCAT folder of the ACS installation (\SPiiPlus MMI Application Studio\Repository Data\EtherCAT), the ACS EtherCAT Configurator tool can be used to configure the network.

If the EtherCAT Configurator tool is already opened, remove it from the workspace and then reload it so it will find the new XML file.

*Note: If an error is generated about non-matching EtherCAT revisions, ACS should be contacted. The low level support algorithm was only tested with EtherCAT revision 0x1.*

* 1. **ACSPL+ Program Setup**

The next part of the setup is configuring the ACSPL+ algorithm. A program file (.prg or .txt) should have been provided with this document that contains the skeleton code for running the low level support algorithm.

The program file can be loaded to the program manager using the program manager’s “Load from File” button. This will load code into buffer 0, buffer 1, and the D-buffer.

*Note: This will overwrite any code located in these 3 buffers. Any code in buffer 0 or buffer 1 should be moved to a different buffer. Any variable declarations in the D-buffer should be copied and added to the D-buffer after the file has been loaded.*

* + 1. **Buffer 0 Setup: EtherCAT Variable Initialization**

Buffer 0 is used for EtherCAT variable initialization. It includes an AUTOEXEC statement that forces this code to be run upon controller restart, as no low level support can be provided without properly initializing the EtherCAT variables.

It should first be decided which ACSPL+ axis will be used for each S700 drive that will be supported.

*Note: Certain limitation on the available axis numbers may exist if other ACS drives or high level supported drives are also setup on the network.*

Once the axis / axes are determined, the subroutine calls for the axis number (MAP\_AXIS$) should be modified to correctly map the inputs (ECIN) and outputs (ECOUT).

The low level support requires the following inputs:

Statusword (SW)

Modes of operation display (MoOD)

Feedback Position (F\_POS)

Position Error (P\_ERR)

The low level support requires the following outputs:

Controlword (CW)

Modes of operation (MoO)

Reference Position (R\_POS)

*Note: Other inputs and outputs may be mapped in this section as well. If the other variables are not used they should be commented out.*

The ACS EtherCAT Configurator tool or the #ETHERCAT terminal command can be used to determine the specific EtherCAT frame address for each of the variables for every S700 drive.

After every required MAP\_AXIS$ routine is configured, the call statements for every required routine should be uncommented, and any unused axis call statements should be commented.

Next, the buffer 1 code should be copied so that every low level supported axis has its own instance of this code (if you have 3 S700 drives this code needs to have 3 copies in 3 different buffers).

Finally, the START buffer commands should be uncommented for each buffer that has an instance of the buffer 1 code.

* + 1. **Buffer 1 Setup: Low Level State Machine**

Next, the code in buffer 1 (at each additional instance of this code) should be configured. This code is the main state machine for every low level supported axis.

Local variables are used to set the ACS axis number (AXIS) and EtherCAT slave ID (SLAVE\_ID). These need to be set appropriately.

The Motion Parameter Initialization section is used to convert encoder counts to user units, as well as to set the initial motion parameters (velocity, acceleration, jerk) in user units. This should be set according to user preference.

The Homing Parameters Initialization section is used to define the homing parameters on the drive for the homing mode of operation (Mode of Operations = 6). This should be set according to user preference.

*Note: The S700 has certain limitations in its homing mode parameters. Kollmorgen should be contacted for any questions in regards to these parameters.*

1. **How to use:**

Once the EtherCAT initialization and state machine for each axis is setup, the drives can be used in almost the same way as a high level supported axis.

* 1. **Enable / Disable**

To enable a drive the ENABLE command can be used.

Example:

ENABLE (1)

To disable a drive the DISABLE command can be used.

Example:

DISABLE (1)

* 1. **Point-to-Point Motion**

To do a point-to-point motion to an absolute position the PTP command can be used

Example:

PTP (1), 0

To do a point-to-point motion to a relative position the PTP/r command can be used

Example:

PTP/r (1), 360

* 1. **Jogging Motion**

To jog the axis at a constant velocity the JOG command can be used

Example:

JOG (1)

* 1. **Homing**

It is recommended to use the drive’s homing methods to home the axis. This is for a number of reasons including the fact that not all the data required for homing is transmitted in real-time (PDOs).

To switch to the homing mode the mode of operations (MoO) variable for the axis should be set to 6.

Note: the state machine automatically disables the axis whenever the mode of operations is changed.

To start homing simply re-enabled the drive. Homing will start using the parameters defined in the Homing Parameters Initialization section. Upon success the mode of operations (MoO) will automatically switch back to 8 and the axis will remain enabled.

Example:

MoO(1) = 6

TILL MoOD(1) = 6

ENABLE (1)

TILL MoOD(1) = 8

* 1. **Fault Clear**

In the current implementation, if the drive is ever in a fault mode it will attempt to clear the fault automatically by toggling the fault clear bit in the controlword (bit 7).

Manual clearing can also be done by toggling this bit.

Example:

CW(1).7 = 1

CW(1).7 = 0

1. **How to Modify the State Machine:**

For safety reasons the state machine code includes a single auto-routine which forces the buffer to restart if for any reason it is ever stopped. With this autoroutine running it is also impossible to make any changes to the buffer code once it starts running.

To get around this the autoroutine for the buffer must be turned off. This can be done by using the program manager, right-clicking on the buffer, and selecting “Disable Autoroutine”. This can also be done by sending the following command through the terminal:

PFLAGS(Buffer#).#NOAUTO = 1.

Once the modifications have been completed the autoroutines should be turned back on. This can be done by using the program manager, right-clicking on the buffer, and selecting “Enable Autoroutine”. This can also be done by sending the following command through the terminal:

PFLAGS(Buffer#).#NOAUTO = 0