PLCs, Software,
Conveyor Controls

JK Bukston, BL. 10
1618 Sofia BULGARIA
Phone: (+359 2) 975 11 80
Fax: (+359 2) 975 11 81
E-mail: indsoft@einet.bg
www.indsoft.bg

Connecting ConveyLinx-Ai/Ai2 modules to Siemens S7 PLCs

Appendix D

Motor Servo Control and Status Description

Rev 1.1

December, 2019

1. Motor Servo Control and Status Description

When a ConveyLinx-Ai2 module is in PLC mode, the user can control the motor movement in **millimeters for a MDR** or in **pulses for a PGD** with the Servo commands. One motor pulse is the change from one hall sensor combination to the next. Formulas below show the connection between number of pulses and the distance in mm. The ConveyLinx-Ai/Ai2 with Senergy-Ai MDR will accept the position value directly in 'mm', as the tube diameter and Gearbox are available to the controller internally.

N of poles – different types of motors have different number of motor poles.

Roller gear ratio – rollers may have different gear ratio, to cover different speed/ torque requirements.

Formula 1:

$$single\ motor\ rotor\ revolution = \frac{N\ of\ poles}{2} * 6(pulses)$$

Formula 2:

$$Roller\ revolution = \frac{rotor\ revolution}{roller\ gear\ ratio}$$

Formula 3:

$$Distance\ in\ mm = \frac{\frac{pulses}{\frac{N\ of\ poles}{2}*6}}{gear\ ratio} * roller\ tube\ diameter(mm) * \pi$$

Example on how to calculate the distance based on formula 3. The example is for Senergy Roller, speed code 60, gear ratio 10.98, tube diameter 48.6 mm. The motor of the Senergy roller has 10 poles. We put as desired distance - 800 pulses. Formula 3 will look this way:

$$Distance\ (mm) = \frac{\frac{800}{10/_2*6}}{10.98} * 48.6 * \pi$$

$$Distance = 370.81\ mm$$

2. Process data control and feedback fields

Output Data - "CLXPLC_OUT"

"**ServoControlDistanceLeft**" – The target left motor position in mm/pulses. The value can be a signed integer value from -32768 to 32767. The maximum traveled distance at once can be 65535 mm/pulses. This is possible in two cases. Case one, if motor is on position 1 and the new motor position is position 5 (see fig. 1). Case two, if motor is on position 5 and the new motor position is position 1.

"**ServoControlDistanceRight**" – The target left motor position in mm/pulses. Same description as for the left motor applies.

The motor will never choose the way of the "overflow". By this we mean, that if the current position is -32000 and the new position is +32000, the motor will rotate for 64 000 mm/pulses. It will not "skip" the route by rotating through the overflow value.

NOTE: If the current motor position is smaller than the new assigned position, the motor will rotate in the CCW direction. If the current position is bigger than the new assigned position, the motor will rotate in the CW direction.

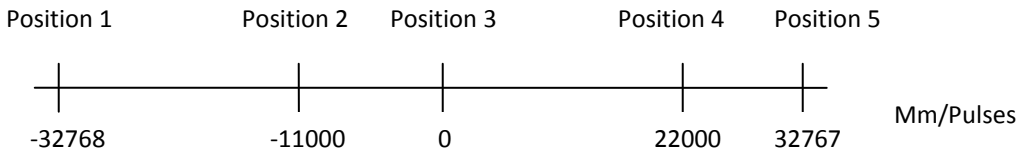

Fig. 1

"**ServoControlCommandLeft**" - Left motor servo commands. Currently only bits 0 and 1 are in use. See fig. 2 for description.

"**ServoControlCommandRight**" – Right motor servo commands. Currently only bits 0 and 1 are in use. See fig. 2 for description.

ServoControlCommandLeft (Word) / ServoControlCommandRight (Word)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														x	x

|	|	
|	|--	Sets the current position as Home position
|--	--	Go to the position

Fig. 2

The two command bits do as follows:

- Sets the current position as Home position. Using this command will set the current position as position 0. The reported position in the Input process data will also report 0 after this command is used.
- Go to position. Setting this command bit will cause the motor to start running towards its target position, which is set in **ServoControlDistanceLeft/ ServoControlDistanceRight**

Input Data – "CLXPLC_IN"

"**DistanceLeft**" – Servo Position Encoder for left motor. This is the current position of the motor. Values can range from -32768 to 32767. When mm/pulses are incrementing motor is rotating in CCW direction. When mm/pulses are decrementing motor is rotating in CW direction.

"**DistanceRight**" – Servo Position Encoder for right motor. See the description for the left motor above.

"**ServoStatusLeft**" – Servo Status for the left motor. Currently only bits 0, 1 and 2 are in use. See fig. 3

"**ServoStatusRight**" – Servo Status for the right motor. See fig. 3.

ServoStatusLeft (Word) / ServoStatusRight (Word)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														x	x

|	|	|	
|	|	|--	Last Command is DONE
|	|--	--	Copy of Home command bit
|--	--	--	Copy of Go to position command bit

Fig. 3

3. Examples

Example 1: Move the left motor from “Position 2” to “Position 4” shown on fig. 2.

Step 1: Current motor position is on position -11000.

Step 2: Set in **ServoControlDistanceLeft** the value of 22000 – this is the position to which, we want to move the motor.

Step 3: Set in **ServoControlCommandLeft** bit 1 – this is the command to move to the position. Motor will rotate CCW.

Step 4: Wait for bit 0 to be “ON” in “**ServoStatusLeft**” – this indicates that the motor has reached the target position. Motor has traveled 33000 mm/pulses from his last position.

Example 2: Move left motor 200 mm/pulses in CW direction.

Step 1: Set bit 0 in **ServoControlCommandLeft** – this will set the current motor position as 0 (home).

Step 2: Wait until you read bit 1 in **ServoStatusLeft** as TRUE, then reset bit 0 in **ServoControlCommandLeft.** Now the position is 0, the home command is no longer active.

Step 3: Wait until you read bit 1 in **ServoStatusLeft** as FALSE. Only now can we continue.

Step 4: Set the value of -200 in **ServoControlDistanceLeft** – this is position we want to move the motor to.

Step 5: Set bit 1 in **ServoControlCommandLeft** – this is the command to move to the position. Motor will rotate CW.

Step 6: Wait for bit 0 in **ServoStatusLeft to** become true – this indicates that the motor has reached the target position.

Notes

a. It is important that the Servo commands are not used alongside the normal Run/Stop commands. These are two separate interfaces to one resource. An example (of a not recommended usage) would be the use of the Run/Stop commands until a product reaches the sensor and then using Servo commands for positioning. If the Run command is retracted in the same process data message, where the Servo command is given, problems may arise. Please contact us in case a mix of Run/Stop commands and Servo commands is necessary for your application. The proper algorithm for these cases is usually application-specific and we can help you to build an issue-free control in this case.
b. If the Servo Command is aborted before the motor has reached its position, the motor will stop with the deceleration ramp.
c. Using Servo brake method is recommended to improve positioning, when using Servo commands. Using normal or free brake methods will lead to positioning errors.

d. If the sum of the Acceleration and Deceleration distances is higher than the travel distance, the module will adjust the Accel/Decel distances to a percentage of the set points for this movement. The set point of the speed may not be reached in this case.