VLC-ETC

USER MANUAL

Version 1.5

SMAC
Moving Coil Actuators

SMAC
Moving Coil Actuators

Disclaimer

The contents of this user manual are intended to be as accurate as possible, but may be subject to change without prior notification. SMAC shall not be liable for any damages that may arise as a consequence of the use of information presented in this user manual.

Document Version	Note	By	Date
1.0	First released version	RZ	7/17/2020
1.1	Upgraded the hardware with the associated pictures, I/O specifications and pinout information.	RZ	11/6/2020
1.2	Updated the firmware version from 1.0.1 to 1.1.0, with the following changes: o new cyclic variables: digital inputs and outputs o new acyclic variables: I2T parameters o fault indication bits in statusword o fault reset through controlword o - modified the system macros	RZ	7/1/2021
1.3	- Fixed the velocity mode system macro, - Updated VLC-ETC housing drawing, added a program example for a 2-axis actuator	RZ	10/28/2021
1.4	- Another fix to the velocity mode system macro, - Added a current sensing offset automatic adjustment in the phasing system macros, - Updated the firmware version from 1.1.0 to 1.1.1, with the following changes: o Controlword bit 2 changed from abort to stop o Macro execution becomes a part of the modes of operation (10) - Added system macros for softland and a PLC program example on how to perform it.	RZ	1/21/2022
1.5	- Fixed the velocity mode system macro - Updated the firmware version from 1.1.1 to 1.1.2, with the following changes: o Fixed the cyclic variable update behaviour during a macro call. o Fixed a glitch observed during successive macro calls	RZ	5/27/2022

SMAC Moving Coil Actuators

Contents

1 Introduction

The VLC-ETC is an EtherCAT servo drive that is based on SMAC's VLC 1-axis integrated controller/driver and an additional layer that provides the EtherCAT connectivity. The VLC part is pre-programmed with system macros to accommodate control and monitoring functionalities of the servo drive. Additional macros can be programmed in the VLC to perform subroutines/functions that can be called from the EtherCAT master. Background knowledge on the VLC is required to configure the servo parameters of the VLC-ETC. Please refer to the latest VLCI-X1 manual for more information about the servo drive parameters and programming. Table 1.1 presents the VLC-ETC specifications.

Table 1.1. VLC-ETC specifications (based on Hardware version 1.0).

Description	EtherCAT servo drive
Operating Modes	Position, Velocity, Torque
Filter Algorithm	PID
Max. Servo Loop Rate	100 µS
Trajectory Generator	Trapezoidal
Servo Position Feedback	Incremental Encoder with Index
Output (Standard)	PWM (space-vector-modulated), 3.5 Amps Cont. and 6.5 Amps Peak (with proper heat mitigation) at 50 VDC Max.
Motor Type	3-Phase Brushless, DC brushed, Linear voice coil
PWM Frequency	20.0 KHz
Current resolution	5.66 mA (approximate)
Encoder and Index Input	Differential
Encoder Supply Voltage	5 VDC
Encoder Input Voltage	5.5 VDC Max., -0.1 VDC Min.
Encoder Count Rate	40 million encoder counts per Second
Position Range	31 Bits
Velocity Range	31 Bits
Acceleration Range	31 Bits
Digital I/O	4x Opto-isolated Digital Inputs w/ common: • 24 V Level Input 4x Solid-state Relay Outputs w/ common: • 200 mA Current • Tolerant to 60 V
Analog I/O	Input: 1 Channel (differential), 0 to +/-10V With 12-Bit Resolution. Output: 1 Channel, 0-10V With 12-Bit Resolution.
STO (Safe Torque Off)	2x Opto-isolated STO Inputs: • 24 V Level Input 1x Opto-isolated STO Feedback Output: • 200 mA Current • Tolerant to 60 V

LEDs	2 x 2 LEDs: • EtherCAT LED: Run (green), Error (red) • Servo Drive LED: Power ON (green), Fault (red)
Communication Interface	• 1x serial/UART (micro USB port): 9600 baud default, selectable between 2400 – 460800 • 2x EtherCAT RJ-45 ports
Supply Voltage	+24 to +48 VDC
Protections	• Driver overtemperature at 150 degrees C • Overcurrent • Overload • Reverse polarity connection • I^2T • (excessive) servo position error
VLC program space	• Macro storage: 53728 bytes • Maximum number of macros: 512 • Maximum number of program registers: 2048

2 Hardware and Software Setup

2.1 Hardware

2.1.1 Power/signal/communication connectors

Figure 2.1 shows the VLC-ETC in its housing. The front and rear sides' connectors as well as LED status indicators are depicted in Figures 2.2 and 2.3. Pinout details of the connectors are presented in the following pages of this manual.

Figure 2.1. VLC-ETC.

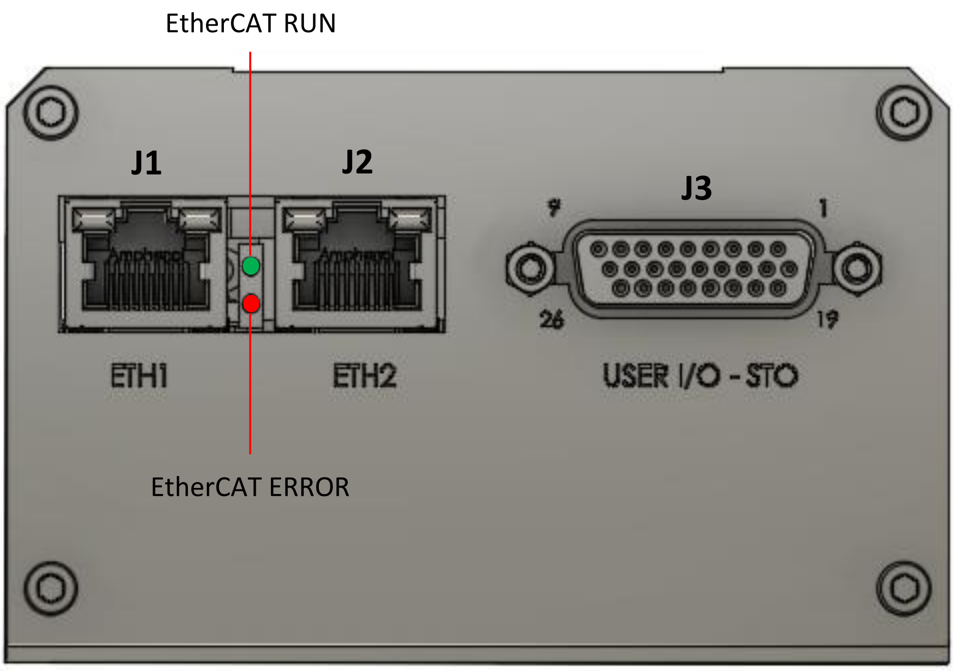


Figure 2.2. VLC-ETC front side connectors and LED status indicators.

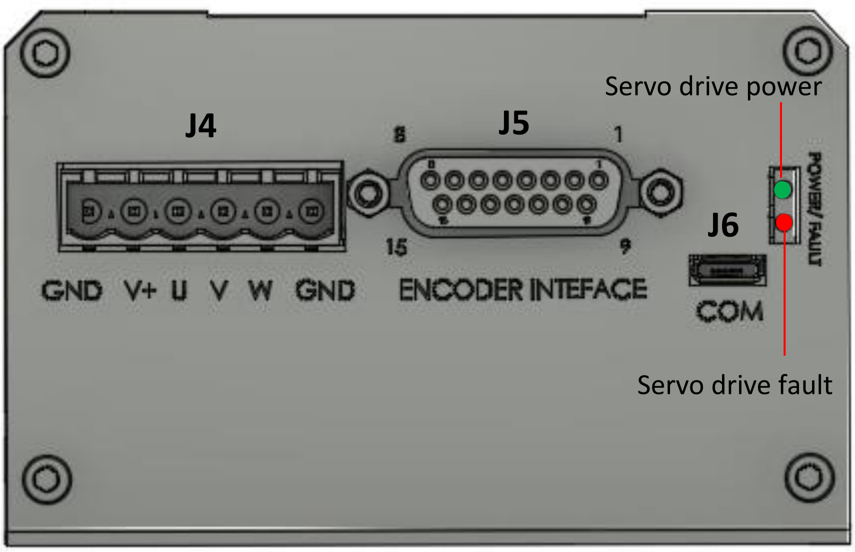
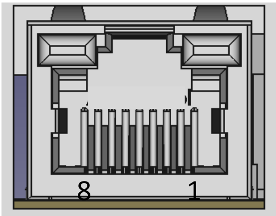


Figure 2.3. VLC-ETC rear side connectors and LED status indicators.

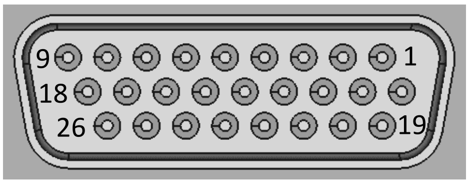
J1 (EtherCAT input) / J2 (EtherCAT output)



RJ-45 Jack.

Pin	Signal	Description
1	TD+	Transmit data +
2	TD-	Transmit data -
3	RD+	Receive data +
6	RD-	Receive data -

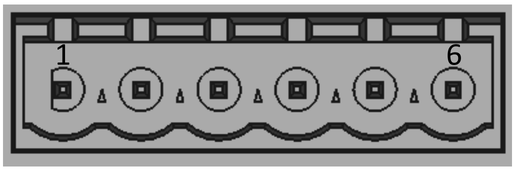
J3 – I/O and STO interface



DSUB26 High Density Female Connector.

Pin number	Signal	Description
1	GPI_COM	Common terminal for general purpose digital inputs
2	GPO_COM	Common terminal for general purpose digital outputs
3	GPI2	General purpose digital input 2
4	GPI0	General purpose digital input 0
5	GND	Ground
6	STO_FB	STO feedback output
7	AN_OUT0	Analog output 0
8	GPO2	General purpose digital output 3
9	GPO0	General purpose digital output 1
10	STO2	STO input 2
11	STO1	STO input 1
12	GPI3	General purpose digital input 2
13	GPI1	General purpose digital input 1
14	GND	Ground
15	NC	Not connected
16	AN_OUT1	Analog output 1
17	GPO3	General purpose digital output 2
18	GPO1	General purpose digital output 0
19 - 22	+5V	+5V power for external circuitry
23	GND	Ground
24	AN_IN0+	Analog input 0 (differential) +
25	AN_IN0-	Analog input 0 (differential) -
26	STO_COM	Common terminal for STO inputs and output

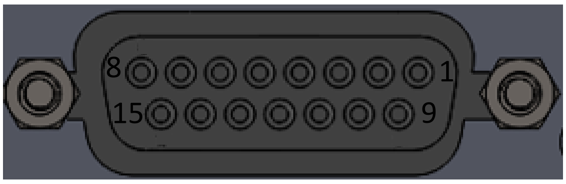
J4 - Power interface



6 pin terminal block header, 5 mm pitch.

Pin number	Signal	Description
1	RTN	Power supply return / ground
2	V+	Power supply positive
3	U	Actuator phase U (positive for single-phase actuators)
4	V	Actuator phase V (negative for single-phase actuators)
5	W	Actuator phase W
6	GND	Ground

J5 – Encoder interface

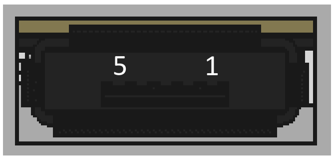


DSUB15 Female Connector.

Pin number	Signal	Description
1	A+	A positive
2	I+	Index positive
3	B+	B positive
4, 5, 6	+5V	+5V supply for encoder
7, 8	NC	Not connected
9	A-	A negative
10	I-	Index negative
11	B-	B negative
12, 13	GND	Ground
14, 15	NC	Not connected

J6 - Serial interface

Note: when this port is connected to a PC, the VLC becomes accessible through a serial terminal software (such as Tera Term) for configuration and programming purposes, while the EtherCAT communication is being interrupted. To operate the VLC-ETC in EtherCAT mode, make sure to first power cycle the VLC-ETC after disconnecting the micro usb cable from this port.

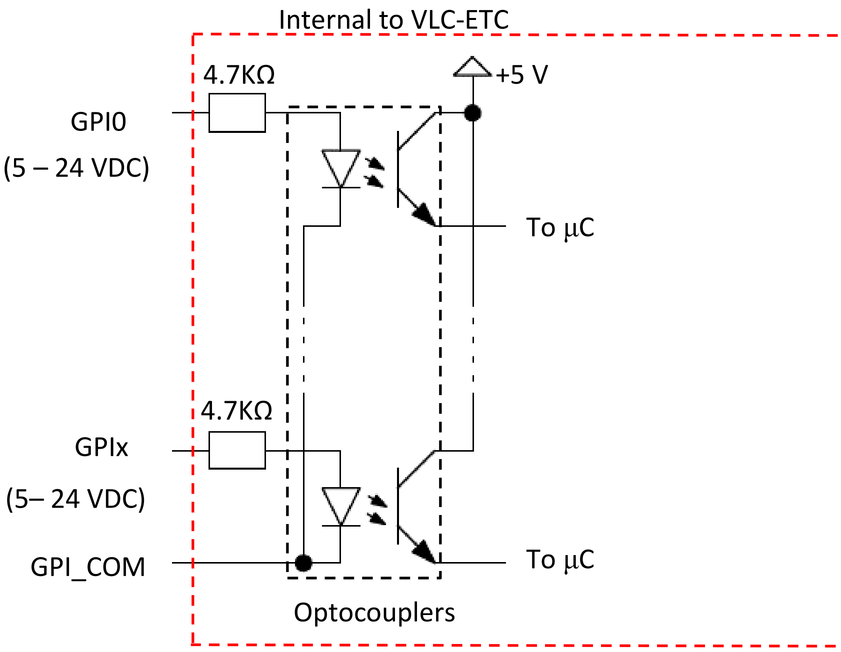


USB Type B (micro) Female Connector.

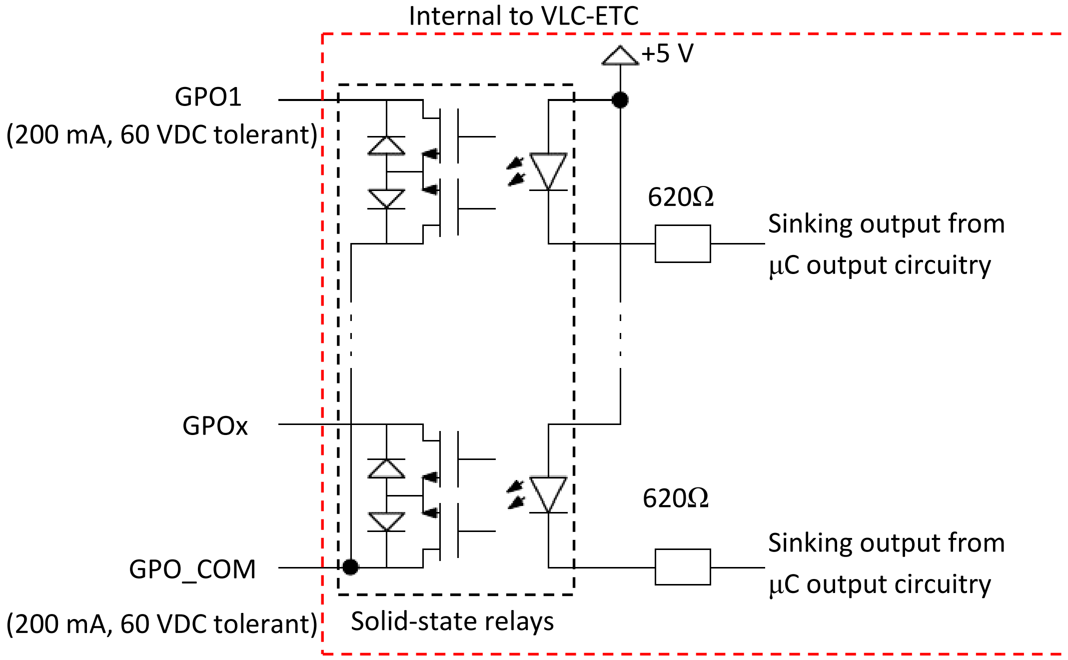
Pin number	Signal	Description
1	USB+	+5 VDC
2	USB D-	Data -
3	USB D+	Data +
4, 5	USB- (GND)	Ground

2.1.2 I/O electrical schematics

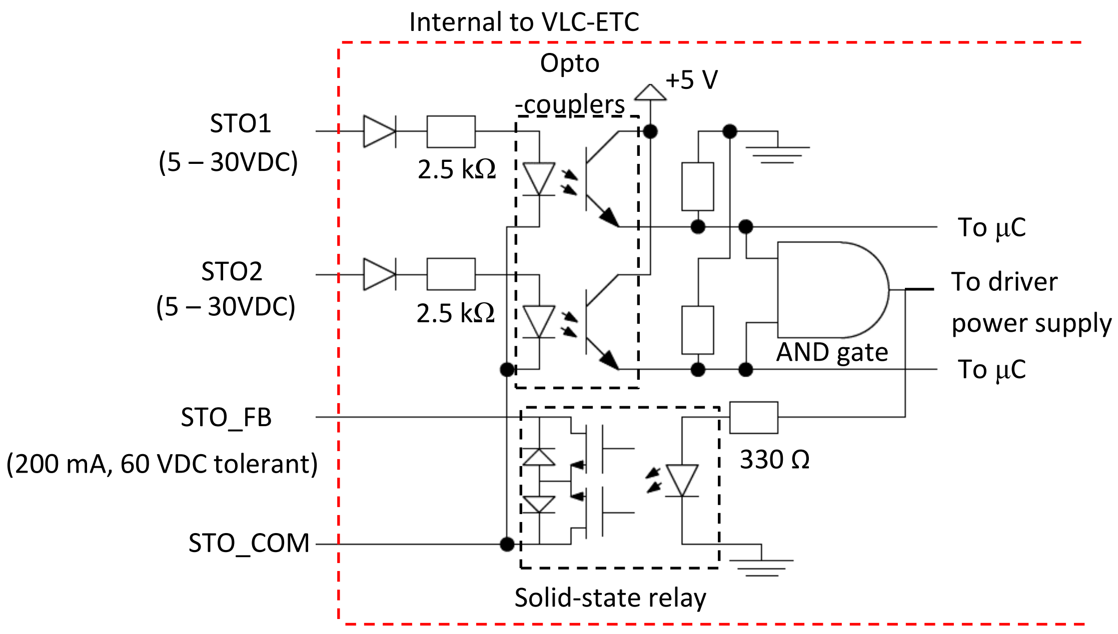
Digital inputs



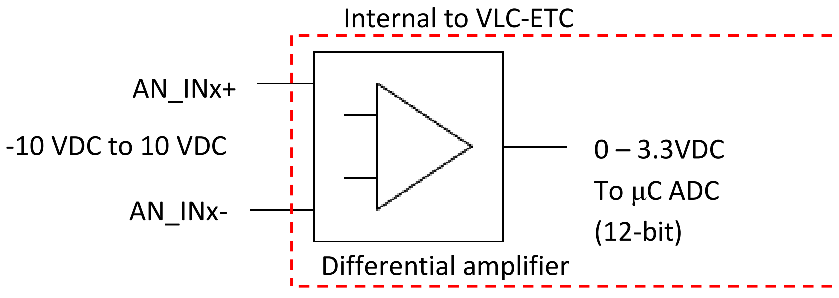
Digital outputs



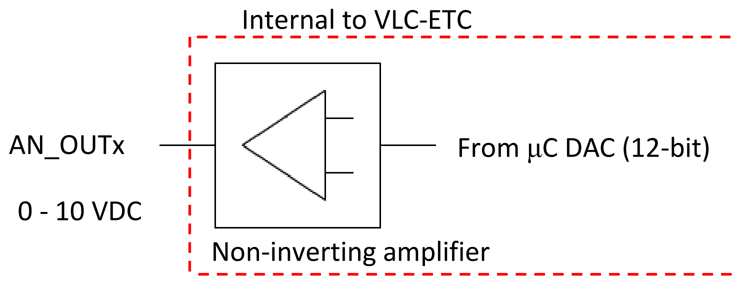
STO



Analog input (differential)



Analog output



2.1.3 Optional: disabling the STO

The two STO inputs (STO1 and STO2, see section 2.1.2) have to be supplied with the specified DC voltage in order to enable the VLCI's driver power stage to operate the actuator. If the external means of supplying the DC voltage is not considered, the VLCI's on-board +5V supply can be used to supply STO1 and STO2, and together with connecting the STO_COM with the GND, the power stage is enabled. This is shown in Figure 2.4. When both STO1 and STO2 are energized, the STO_FB output becomes active to indicate the drive is ready to be operated.

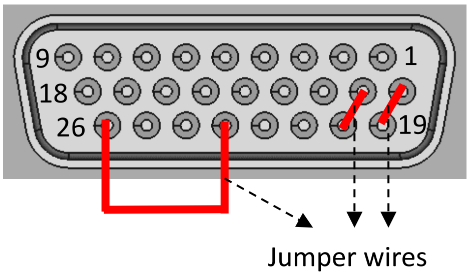


Figure 2.4. Disabling STO with the on-board +5V supply.

2.2 Software Setup

2.2.1 VLC configuration

Remark: to perform VLC configuration, the user is expected to be familiar with programming the VLC/LAC. Refer to the latest VLCI-X1 manual for more information on the programming.

Serial communication can be established between the VLC part and a PC through the micro USB port (J6) of the VLC-ETC. A serial terminal software (with selectable baud rates of up to 460800, such as Tera Term) can be used to configure the VLC for the following purposes:

- Loading of system macros (see appendix A, unless these macros were pre-loaded in the VLC).
- Setting/tuning of actuator servo parameters (Optional, as this can also be done through EtherCAT). The servo parameters can later be stored in MD151-MD154 presented in appendix A, which later will be loaded onto the EtherCAT side upon VLC-ETC power-up (after a PS command).
- Programming of custom macros that are to be called via EtherCAT (Optional).

Appendix A presents the system macros. Due to these macros, there are restrictions in programming the custom macros as follow:

- All macros can be used except: 0, 151-159, 200-255
- All registers can be used except: 200-229, 400-403

For the VLC configuration, the baud rate setting of the terminal software has to be adjusted to 460800. Also note that if a serial echo is desired, this can be enabled through the `EN` command.

2.2.2 Connecting the VLC-ETC to an EtherCAT network (with a TwinCAT example)

The VLC-ETC is ready to be connected to an EtherCAT network, provided the following conditions are met:

- System macros have been loaded and saved in the VLC.
- The micro USB cable is disconnected from the USB port (J6) and after that, the VLC-ETC is power-cycled.

2.2.2.1 EtherCAT master cycle time setting

The cycle time determines the update rate of EtherCAT slave data. For VLC-ETCs, generally the minimum cycle time is limited to 2 milliseconds.

2.2.2.2 Connecting the VLC-ETC to TwinCAT software

The ESI (EtherCAT Slave Information) file of the VLC-ETC has to be copied into the following typical directory in a PC with TwinCAT installed: C:\TwinCAT\3.1\Config\Io\EtherCAT . The ESI file can be obtained from SMAC.

The following are the steps to establish a connection between VLC-ETC and TwinCAT:

- Open TwinCAT and start a new TwinCAT project
- Ethernet adapter installation: On the top part of the development environment (Fig. 5), select TwinCAT > Show Realtime Ethernet Compatible Devices. Make sure that the Ethernet Adapter of the PC is installed.
- On the left pane of the TwinCAT development environment, right-click on I/O>Devices, select "Add new item". A window "Insert Device" appears as shown in Figure 2.5. Select EtherCAT Master and click OK.
- A window "Device Found At" appears as in Figure 2.6, select the Ethernet adapter that is already installed.

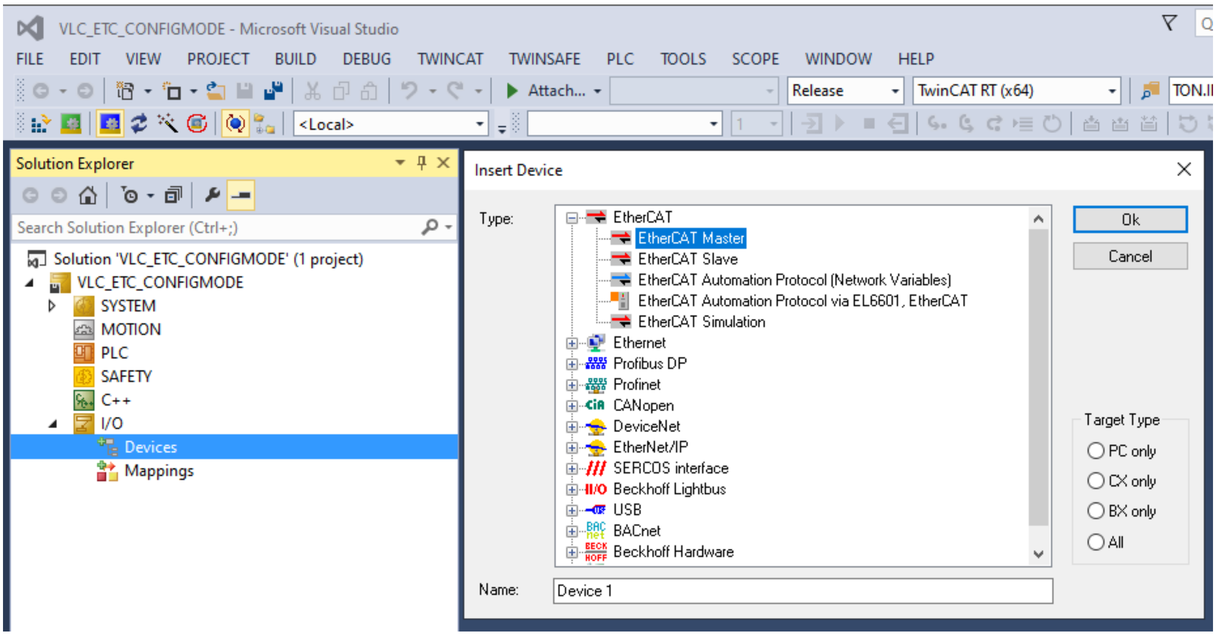


Figure 2.5. Adding an EtherCAT master.

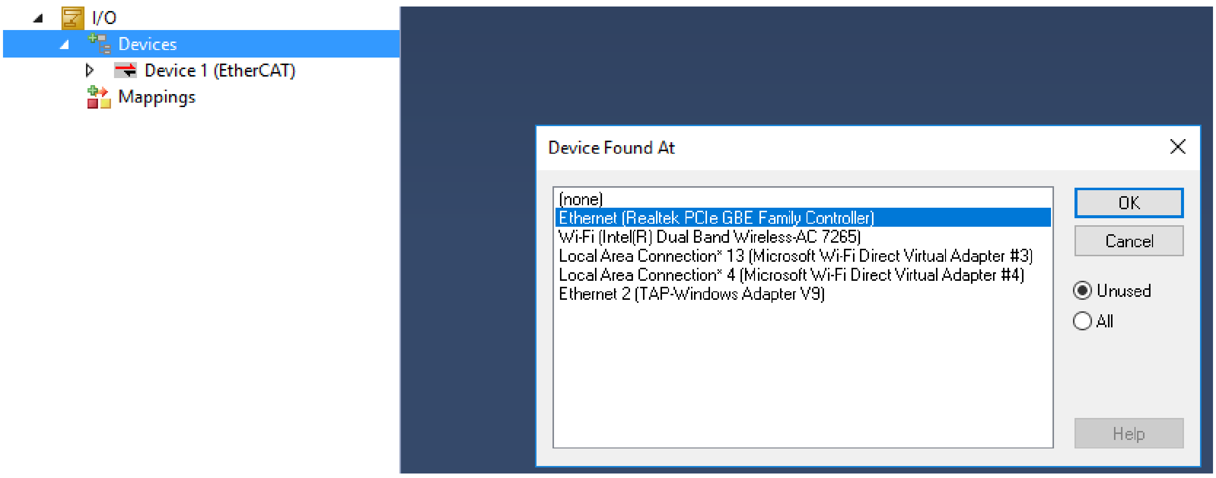


Figure 2.6. Selection of an Ethernet adapter.

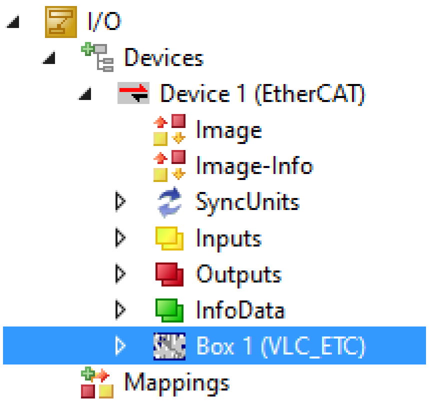


Figure 2.7. VLC-ETC connected to TwinCAT.

➢ Right-click on the newly-added "Device 1 (EtherCAT)" and select scan. The VLC-ETC will appear as shown in Figure 2.7.

➢ To view the various data belonging to VLC_ETC, double-click on Box 1 (VLC_ETC). A list of data will appear as shown in Figure 2.8. The data in the upper dashed rectangle are configuration objects, which are acyclic (only updated upon request). Meanwhile, the lower dashed rectangle contains cyclic objects (updated periodically) in the EtherCAT OPERATIONAL state.

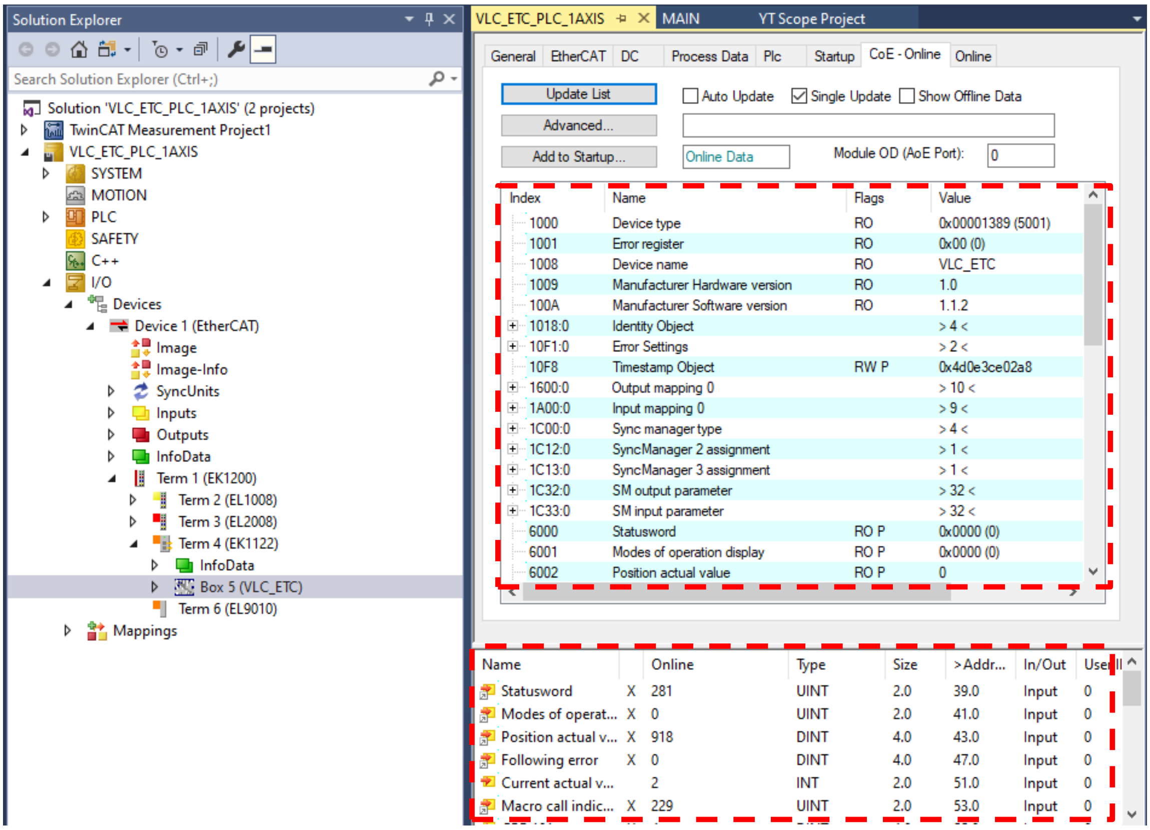


Figure 2.8. Various acyclic (upper rectangle) and cyclic (lower rectangle) objects of VLC_ETC.

3 Programming the VLC-ETC

3.1 Servo objects

Servo objects are parameters and variables that are used to perform control and monitoring of the VLC-ETC. From its update behaviour perspective, there are two servo object types in VLC-ETC:

- Acyclic: updated upon request, used for servo configuration purposes.
- Cyclic: updated periodically, manipulated and monitored by a motion program executed by the EtherCAT master.

3.1.1 Acyclic servo objects

Table 3.1 presents the acyclic servo objects. Objects with indices 8000, 8001, 8002, 8004 (partially), 8005 and 8006 are standard ones that are also found in VLC, therefore, explanations on their usage are available in the VLC user manual. Additionally, commutation electrical cycle (index: 8004) corresponds to the `EC` command in VLC, which is used for phasing of 3-phase actuators.

Table 3.1. List of acyclic servo objects.

Index	Sub-index	Object	Data Type	VLC equivalent command/variable
8000		POSITION LOOP CONTROLLER PARAMETERS		
	1	Proportional constant	UINT16	`SG`
	2	Integral constant	UINT16	`SI`
	3	Derivative constant	UINT16	`SD`
	4	Integral limit	UINT16	`IL`
	5	Velocity feedforward constant	UINT16	`FV`
	6	Acceleration feedforward constant	UINT16	`FA`
	7	Derivative sampling frequency	UINT8	`FR`
	8	Integral Sampling gain	UINT8	`RI`
8001		Current mode gain	UINT16	`SC`
8002		ADDITIONAL SERVO PARAMETERS		
	1	Servo speed	UINT8	`SS`
	2	Phase and sense setting	UINT8	`PH`
	3	Dead band	UINT16	`DB`
	4	Output offset	INT16	`OO`
	5	Maximum following error	UINT16	`SE`
8003		HOMING PARAMETERS		
	1	Homing method	UINT16	N/A
	2	Homing speed	UINT32	N/A
	3	Homing acceleration	UINT32	N/A
	4	Home offset	INT32	N/A
	5	Position error threshold	UINT32	N/A
	6	Homing timeout	UINT16	N/A

Index	Sub-index	Object	Data Type	VLC equivalent command/variable
8004		COMMUTATION PARAMETERS		
	1	Commutation phase	UINT16	SP
	2	Commutation electrical cycle	UINT32	EC
	3	Commutation voltage	UINT16	N/A
	4	Absolute home	INT32	DA
8005		GENERAL PURPOSE REGISTERS		
	1	GPR11	INT32	AR11
	2	GPR12	INT32	AR12
	…			
	15	GPR25	INT32	AR25
8006		I2T PARAMETERS		
	1	I2T Nominal current	UINT16	i2t_NOM
	2	I2T Trip level	UINT32	i2t_TRIP
8007		Save all parameters	UINT8	PS

Important notes (for more information, see the program examples in the VLCI-X1 manual):

- **For 1-phase actuators, set the object "Commutation phase" (8004;1) to 27307**
- **For 3-phase actuators, set the object "Phase and sense setting" (8002;2) to 1**

3.1.1.1 Homing parameters

Homing parameters (index: 8003) are specific to the VLC-ETC. To perform homing through the manipulation of cyclic objects (further description in the following subsection of this manual), phasing (in case a 3-phase actuator is used) has to be executed successfully beforehand. Furthermore, position loop control parameters have to be set properly since homing involves a controlled motion. Therefore, in principle all the relevant objects in Table 3.1 will have to be set prior to homing. The following objects describe the homing parameters of Table 3.1 in detail:

- Homing method (value depends on the chosen method below)
  - Current position (0): no motion is involved. This sets the position value = "Home offset" object.
  - Negative mechanical limit (1): retracts the shaft until actuator the rear bumper is detected, and sets the position value = "Home offset" object.
  - Positive mechanical limit (2): extends the shaft until the actuator front bumper is detected, and sets the position value = "Home offset" object.
  - Negative index (3): retracts the shaft until the index is detected.
  - Positive index (4): extends the shaft until the index is detected.
  - Negative mechanical limit and index (5): retracts the shaft until actuator the rear bumper is detected, extends until the index is detected and sets the position value = "Home offset" object
  - Positive mechanical limit and index (6): extends the shaft until actuator the rear bumper is detected, retracts until the index is detected and sets the position value = "Home offset" object

- Homing speed: speed of retracting and extending shaft movements during homing.
- Homing acceleration: acceleration of the shaft movement during homing.
- Home offset: the value sets to the actuator's actual position after homing is completed.
- Position error threshold: the position error value to conclude the existence of the mechanical limit.
- Homing timeout: the time period allowed to complete the homing. If homing has not been completed within the time period, homing is considered to be failed.

3.1.1.2 Commutation parameters

Except for the commutation voltage, the rest of commutation parameters are available in the standard VLC. The relevance of these parameters for the actuator operation are as follow:

- 1-phase (brushed/voice coil) motor: Commutation phase has to be set to 27307. See the VLC manual for more information on this object.
- 3-phase (brushless) motor: Commutation voltage and electrical cycles are to be configured to perform phasing. Additionally, the object Phase and sense setting (8002;2) have to be set to 1. See the program example in VLC manual for more information.

3.1.1.3 I2T parameters

These parameters are used to configure the actuator overloading characteristic in terms of peak current and time period, which are intended to protect the actuator from possible overheating. A further discussion on this topic is covered in the VLCI-X1 manual.

3.1.1.4 Save all parameters

By changing the value of the object Save all parameters (index: 8007) from 0 to 1 or vice-versa, all acyclic servo objects from index 8000 – 8006 will be saved to the non-volatile memory of the VLC, which allow the object values to be retained after a VLC-ETC power cycle.

3.1.2 Cyclic servo objects

In Table 3.2, the **INPUT** and **OUTPUT** are seen from the EtherCAT master / TwinCAT perspective. Some of the objects have their VLC equivalence and therefore, further information about them are to be found from the VLC manual. The **OUTPUT** object value is applied to the VLC-ETC upon <u>a change of its value</u>. Note that GPR's 101 – 104 are only meaningful when they are used within custom macros that can be called by the EtherCAT master.

SMAC Moving Coil Actuators

Table 3.2. List of cyclic servo objects.

I/O	Object	Data type	VLC equivalence
INPUT	Statusword	UINT16	N/A
	Modes of operation display	UINT16	N/A
	Position actual value	INT32	`TP` command
	Following error	INT32	`TF` command
	Current value	INT16	`IMON` variable
	Macro call indicator	UINT16	N/A
	GPR 101	INT32	Register 101
	GPR 102	INT32	Register 102
	Digital inputs	UINT8	`BI` command
OUTPUT	Controlword	UINT16	N/A
	Modes of operation	UINT16	N/A
	Setpoint	INT32	N/A
	Profile velocity	UINT32	`SV` command
	Profile acceleration	UINT32	`SA` command
	Maximum torque	INT16	`SQ` command
	Macro call	UINT16	`MS` command
	GPR 103	INT32	Register 103
	GPR 104	INT32	Register 104
	Digital outputs	UINT8	`BO` command

Meanwhile, for objects that do not have a direct VLC equivalence:

3.1.2.1 Statusword

The statusword contains bits with various servo status, as described in Table 3.3.

Table 3.3. Statusword bits.

Bit	Description
0	**Initialization done.** This will be set to 1 after VLC-ETC performs its initialization process upon power-up, indicating that it is ready to be operated.
1	**Servo enabled.** This will be set to 1 when the servo is enabled by `MN` command. The bit is set to 0 if servo is disabled through `MF` command.
2	**Reserved.**
3	**Motion execution acknowledge bit.** Set to 1 on a rising edge transition the controlword “start motion” bit and set to 0 on falling edge transition of the same controlword bit.
4	**Trajectory complete.** Set to 1 if the servo has completed a position move. Set to 0 if the servo is busy executing a commanded move.
5	**Reserved.**
6	**Homing Success.** Set to 1 after homing has been completed successfully.
7	**Homing failure.** Set to 1 if homing fails (when the homing process takes longer than the configured timeout value).
8	**Phasing success.** Set to 1 after phasing is successfully performed. Set to 0 upon failed phasing or at power-up.
9	**Phasing failure.** Set to 1 after a failed phasing. Set to 0 upon successful phasing or at power-up. A failed phasing could be caused by incorrect “commutation electrical cycle”, incorrect “phase and sense setting” or insufficient phasing setpoint value.

10	**Macro execution error.** Set to 1 if an undefined macro is called. Set to 0 when a new call to a defined macro is performed, or any of the mode of operation is executed.
11	**Macro execution.** Set to 1 when a macro is being executed. Set to 0 when macro execution is completed.
12	**General fault.** Set to 1 in the event of overtemperature, I2T being tripped, or STO is activated. Set to 0 otherwise, or if the fault has been successfully reset.
13	**Servo error.** Set to 1 when the following error has exceeded the acyclic variable "Maximum following error". Set to 0 otherwise, or if the fault reset has been executed.
14	**Reserved.**
15	**STO status**. Set to 1 when STO is active (or if any of the STO inputs is not energized). Set to 0 when STO is inactive.

3.1.2.2 Controlword

The controlword contains bits that represent certain servo functions, as described in Table 3.4.

Important notes:

- **Only one of the bits between 0 – 5 and 9 can have a value of 1 at the same time.**
- **By setting all bit values to 0 (same as decimal value: 0), macro execution will be stopped and servo is turned off.**

Table 3.4. Controlword bits.

Bit	**Description**
0	**Idle.** This bit does not do anything, but can be used to allow an easy transition of controlword decimal value. Example of use is given in the upcoming section.
1	**Start motion/macro call.** On a rising edge, starts the execution of motion or macro call based on the selected mode of operation.
2*	**Stop motion.** On a rising edge, motion is stopped (for modes of operation 1, 2, 3) and servo is held in its position.
3	**Motor off.** On a rising edge, servo is turned off.
4	**Motor on.** On a rising edge, servo is enabled.
5	**Abort macro execution.** On a rising edge, macro execution is aborted.
6 - 8	**Reserved.**
9	**Fault reset.**
10 - 15	**Reserved.**

* Prior to firmware version 1.1.1, this bit performs an 'abort motion' operation.

3.1.2.3 Modes of operation, modes of operation display, setpoint, macro call indicator

There are seven servo modes of operation, for which the object "setpoint" has a dependent function, as described in table 3.5.

Table 3.5. Modes of operation.

Modes of operation	**Value**	**Description**	**Setpoint function**
Absolute position move	1	Equivalent to `PM,MA` in VLC.	Absolute target position value
Relative position move	2	Equivalent to `PM,MR` in VLC.	Relative target position value
Velocity move	3	Equivalent to `VM` in VLC.	Target velocity value. Can be either positive or negative, unlike in the case of `VM` in VLC, where direction has to be set through `DI`.
Torque move	4	Equivalent to `QM0` in VLC. Drives the actuator through an open loop voltage command.	Same as `SQ` value in `QM` mode.
Current move	5	Equivalent to `QM1` in VLC. Drives the actuator through a closed loop current command.	Same as `SQ` value in `QM1` mode.
Homing	6	Performs homing. See subsection 3.1.1.1 for more information.	N/A
Phasing	7	Performs phasing for 3-phase (brushless) actuators/motors. See subsection 3.1.1.2 for more information.	N/A
Macro execution**	10	Performs macro execution based on the "macro call" cyclic object value	N/A

The object "modes of operation display" follows the "modes of operation" value, such mechanism can be used as a handshake for control purposes.
The object "Macro call indicator" shows the macro number being executed during a servo move and during the execution of a macro based on the "macro call" object.
** Prior to firmware version 1.1.1, this mode of operation does not exist, so macro call would be executed when a new "macro call" value (which has been pre-programmed into the VLC-ETC) is written by the EtherCAT master. This would be ineffective when the macro call is to be executed repeatedly as it would require a new "macro call" value to be written every time. Starting from firmware version 1.1.1, this has been addressed, by including a macro execution mode of operation and therefore, the actual macro call can be executed simply by changing the controlword value as explained in the following section.

3.2 Executing motion through servo objects

The following are the steps to execute motion corresponding to one of the modes of operation:

- Configure the acyclic servo objects (Table 2) that are necessary for the selected mode of operation.
- Set the object "Modes of operation" to the desired value.
- Set the cyclic output objects required for the modes of operation as desired, for instance, to execute a position move, "Profile velocity", "Profile acceleration" and "Setpoint" need to be set to the desired values.
- Set bit 1 of controlword to 1. This will start the motion.
- Bit 3 of statusword will be set to 1 after the above to indicate the motion has been started.
- Set bit 1 of controlword back to 0, this will also be responded by bit 3 of statusword accordingly, as shown in Figure 3.1.

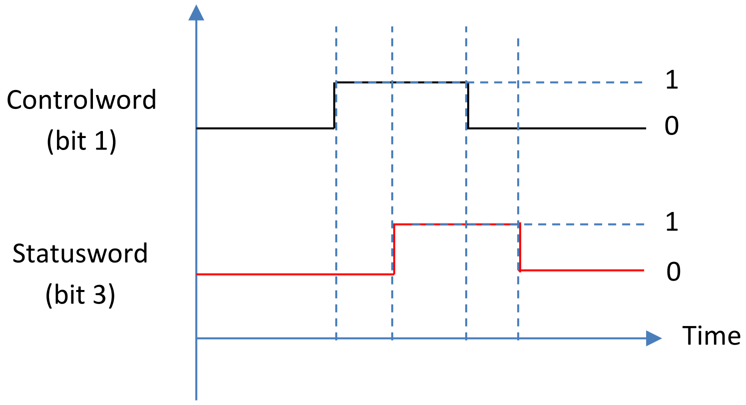


Figure 3.1. Controlword bit 1 and statusword bit 3 handshake mechanism.

3.3 Example: executing homing with TwinCAT under Config mode

- Click the icon in the dashed rectangle shown In Figure 3.2 to reload EtherCAT devices and to activate the free-run state of VLC-ETC. Once this is done, the green LED next to the EtherCAT port should stay on.
- Perform the configuration of acyclic servo objects relevant for homing, which include PID constants and homing parameters. This is done by double-clicking the objects from the list in Figure 3.3 and set the values as desired. After this is done on all parameters, one could also save the parameters into the non-volatile memory by setting the "save all parameters" object (8007) to 1.

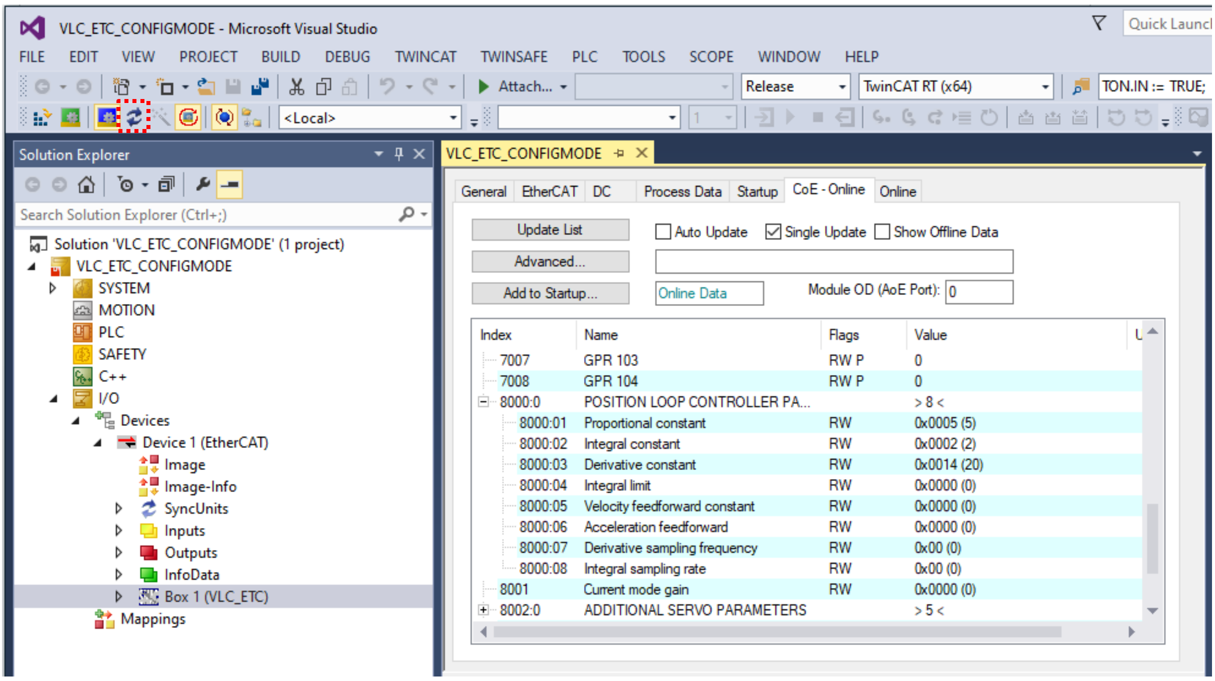


Figure 3.2. Acyclic servo objects list.

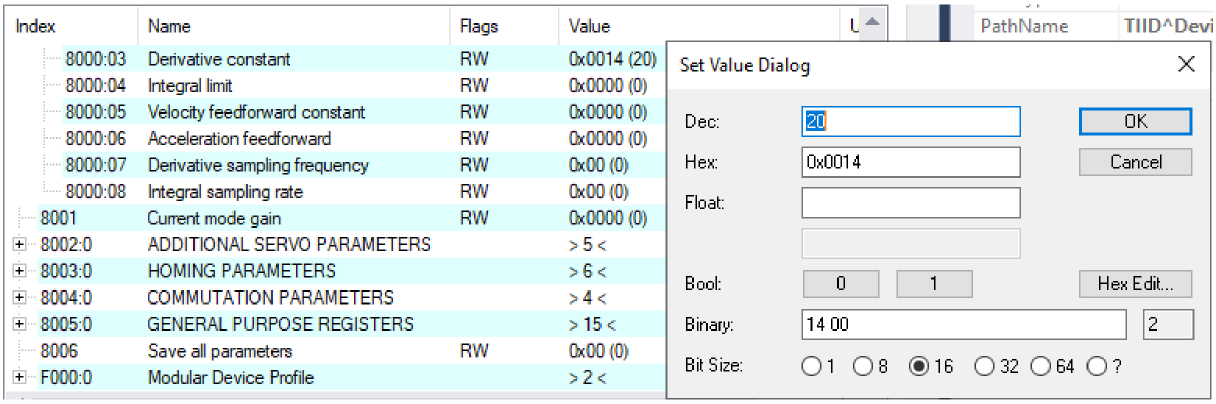


Figure 3.3. Setting the value of an acyclic servo object.

- Under the acyclic servo objects list, there is a list of cyclic object list. Scroll down to find the object “Modes of operation”. Right-click and set its value to 6.
- Next, set the object “controlword” value to 2. This will start the homing motion.
- After homing is completed, either bit 6 or 7 of statusword will be set to 1, as shown in Figure 3.4. Use a decimal to binary converter to identify this.
- Set bit 2 of controlword to 0, this can be done by setting the decimal value of controlword to 1 as shown in Figure 3.5, instead of setting it to 0 which in some cases would terminate macro execution and turn off the servo. This change will set the statusword motion execution acknowledge (bit 3) to 0.

Name	Online	Type	Size	>Addr...	In/Out	User ID	Linked to
Statusword	347	UINT	2.0	39.0	Input	0	
Modes of operat...	6	UINT	2.0	41.0	Input	0	
Position actual v...	13	DINT	4.0	43.0	Input	0	
Following error	-15	DINT	4.0	47.0	Input	0	
Current actual v...	-16	INT	2.0	51.0	Input	0	
Macro call indic...	229	UINT	2.0	53.0	Input	0	
GPR 101	0	DINT	4.0	55.0	Input	0	
GPR 102	0	DINT	4.0	59.0	Input	0	
WcState	0	BIT	0.1	1522.1	Input	0	
InputToggle	0	BIT	0.1	1524.1	Input	0	
State	8	UINT	2.0	1548.0	Input	0	
AdsAddr	192.168.0.110.2.1:1...	AMSADDR	8.0	1550.0	Input	0	
Controlword	2	UINT	2.0	39.0	Output	0	
Modes of operat...	6	UINT	2.0	41.0	Output	0	
Setpoint	0	DINT	4.0	43.0	Output	0	

Figure 3.4. Statusword value with bit 6 set to 1, indicating homing has been successfully completed.

Name	Online	Type	Size	>Addr...	In/Out	User ID
Statusword	339	UINT	2.0	39.0	Input	0
Modes of operat...	6	UINT	2.0	41.0	Input	0
Position actual v...	13	DINT	4.0	43.0	Input	0
Following error	-15	DINT	4.0	47.0	Input	0
Current actual v...	-23	INT	2.0	51.0	Input	0
Macro call indic...	229	UINT	2.0	53.0	Input	0
GPR 101	0	DINT	4.0	55.0	Input	0
GPR 102	0	DINT	4.0	59.0	Input	0
WcState	0	BIT	0.1	1522.1	Input	0
InputToggle	0	BIT	0.1	1524.1	Input	0
State	8	UINT	2.0	1548.0	Input	0
AdsAddr	192.168.0.110.2.1:1...	AMSADDR	8.0	1550.0	Input	0
Controlword	1	UINT	2.0	39.0	Output	0

Figure 3.5. Setting bit 2 of controlword back to 0.

3.4 Example: programming a sequence of motions with TwinCAT PLC under Run mode

- Start a new TwinCAT project, select the target system according to the name of the Beckhoff embedded/industrial PC being connected in rectangle a in Figure 3.6. More information on how to make the connection between TwinCAT and the embedded PC is in the Beckhoff's manual. Also, in rectangle b, select TwinCAT version corresponding to the embedded PC being used.
- Right click on I/O>Devices and select Scan. A new window will appear showing the I/O devices being connected. Select 'Device 1'.
- Another window will appear asking to scan for 'boxes'. Click OK and EtherCAT terminals will appear and VLC-ETC will be under one of them as shown in Figure 3.7 (provided the VLC-ETC is connected to the EtherCAT junction terminal).
- Right-click on the PLC in the left pane of TwinCAT environment and select 'Add new item'. Provide a name for a standard PLC project, as illustrated in Figure 3.8.

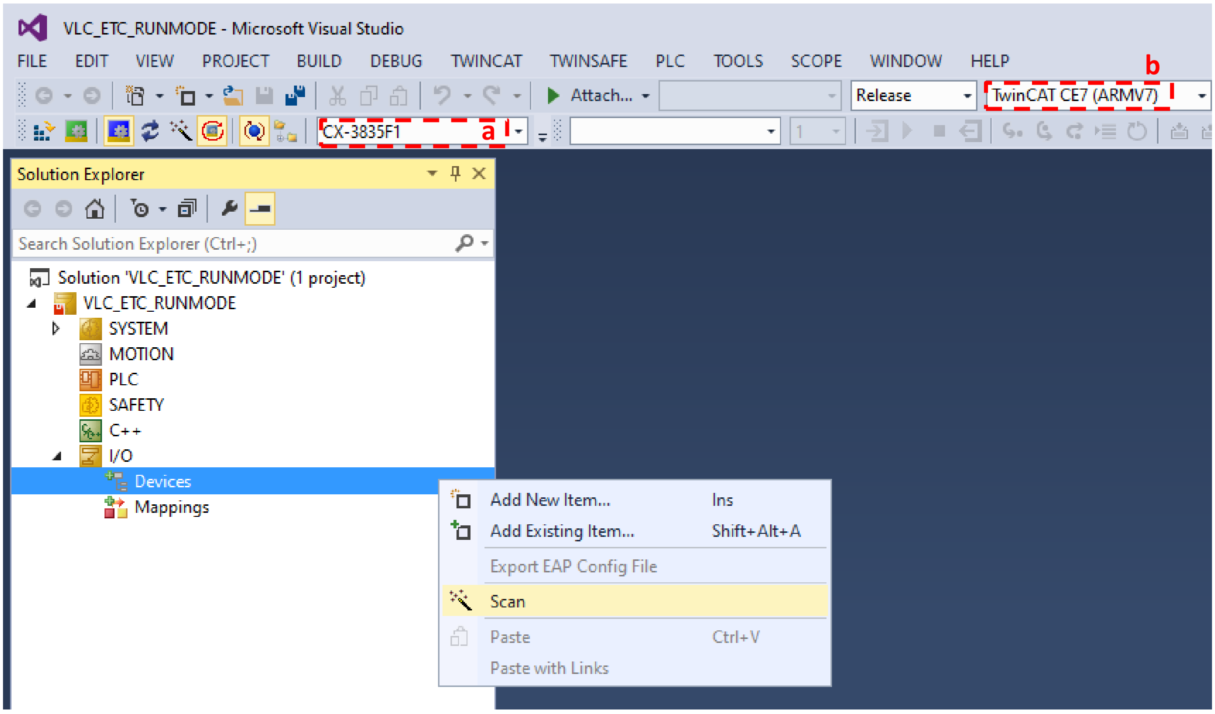


Figure 3.6.

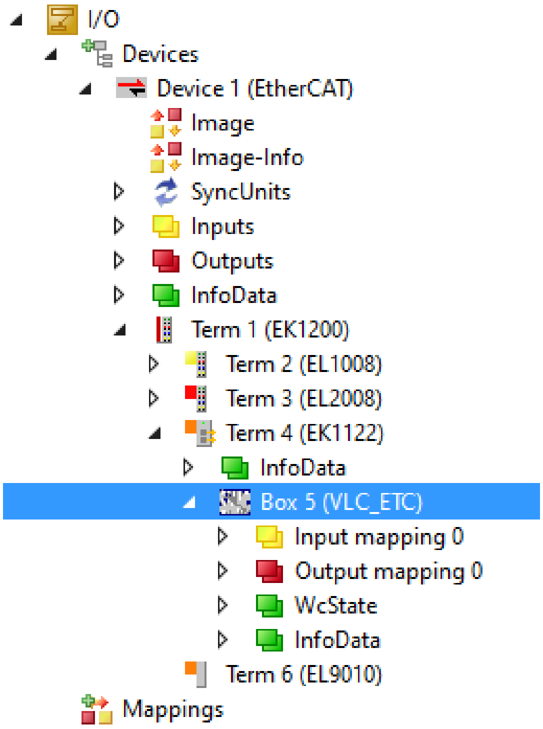


Figure 3.7. VLC-ETC under one of the scanned terminals.

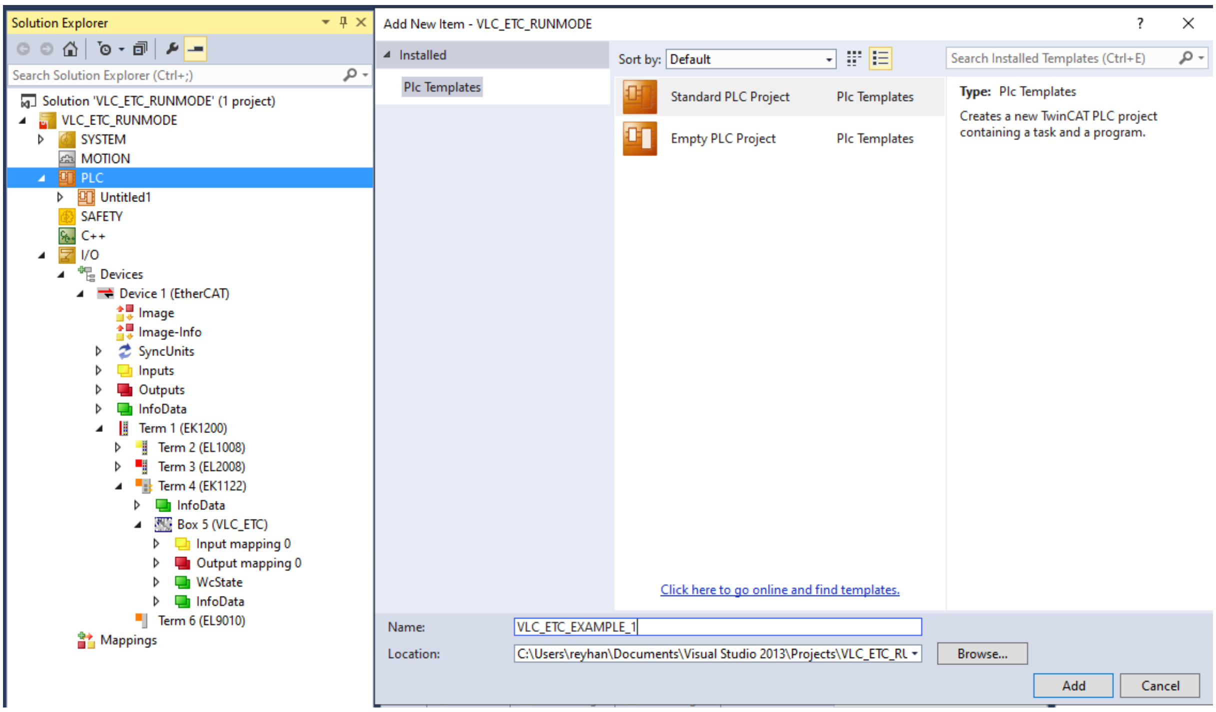


Figure 3.8. Adding a new PLC project.

- Go to the PLC program editor as shown in the left pane of Figure 3.9. The top rectangle is where the program variables are declared, while the bottom one is where the program logic will be located. In this example, the sequence of motions consists of: phasing – homing – repetitive position move.

    - Enter the following piece of code in the top rectangle (note that some of the variables are only relevant in the context of example 3.5 in this manual):

```
PROGRAM MAIN
VAR

// declaration of input variables
Statusword AT%I* : UINT;
Pos_value AT%I* : DINT;
Mode_Op_Disp AT%I* : UINT;
Foll_error AT%I* : DINT;
GPR101 AT%I* : DINT;

// declaration of output variables
Controlword AT%Q* : UINT;
Mode_Op AT%Q* : UINT;
Setpoint AT%Q* : DINT;
Vel AT%Q* : DINT;
Acc AT%Q* : DINT;
Macro_call AT%Q* : UINT;
GPR103 AT%Q* : DINT;
GPR104 AT%Q* : DINT;

// switch-case state
State : INT :=0;

// phasing error indicator
Phasing_Error : BOOL := FALSE;
Homing_Error : BOOL := FALSE;

END_VAR
```

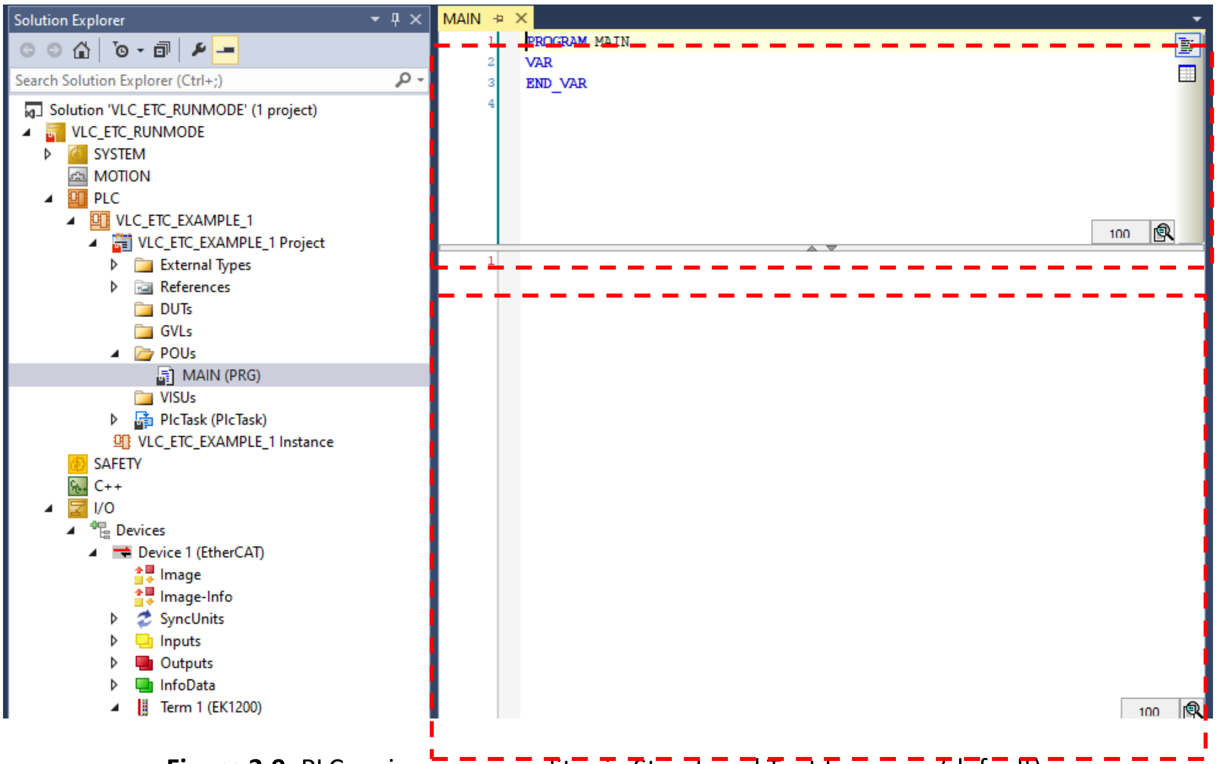


Figure 3.9. PLC main program editor in Structured Text language (default).

- Enter the following piece of code in the bottom rectangle:

```
CASE State OF

0: // check if the drive is ready to be operated, statusword bit 0 = 1
IF (Statusword.0)=1 THEN
State := 1;
END_IF

1: // change mode of operation to phasing
Mode_Op := 7;
State := 2;

2: // check if mode of operation has been set accordingly
IF Mode_Op_Disp = Mode_Op THEN
State := 3;
END_IF

3: // start phasing
Controlword := 2;
State := 4;
```

```
4: // Check if Phasing has been completed
IF (Statusword.8) =1 THEN
Controlword := 1;
Phasing_Error := FALSE;
State := 5;
END_IF

IF (Statusword.9) =1 THEN
Controlword := 1;
Phasing_Error := TRUE;
END_IF

5: // change mode of operation to homing
Mode_Op := 6;
State := 6;

6: // check if mode of operation has been set accordingly
IF Mode_Op_Disp = Mode_Op THEN
State := 7;
END_IF

7: // start homing
Controlword := 2;
State := 8;

8: // Check if Homing has been completed
IF (Statusword.6) =1 THEN
Controlword := 1;
Homing_Error := FALSE;
State := 9;
END_IF

IF (Statusword.7) =1 THEN
Controlword := 1;
Homing_Error := TRUE;
END_IF

9: // change mode of operation to position move (absolute)
Mode_Op := 1;
State := 10;

10: // check if mode of operation has been changed
IF Mode_Op_Disp = Mode_Op THEN
State := 11;
END_IF
```

```
11: // Enter motion parameters
Vel := 100000;
Acc := 1000000;
Setpoint := 4000;
Controlword := 2;
State := 12;

12: // Check if the movement has started
//IF (Statusword.3)=1 THEN
IF (Statusword.4)=0 AND (Statusword.3)=1 THEN
Controlword := 1;
State := 13;
END_IF

13: // check if target position has been (almost) reached
//IF ABS(Setpoint-Pos_value)<100 THEN
IF (Statusword.4)=1 THEN
State := 14;
END_IF

14: // Enter motion parameters
Vel := 1000000;
Acc := 10000000;
Setpoint := 1000;
Controlword := 2;
State := 15;

15: // Check if the movement has started
//IF (Statusword.3)=1 THEN
IF (Statusword.4)=0 AND (Statusword.3)=1 THEN
Controlword := 1;
State := 16;
END_IF

16: // check if target position has been (almost) reached
//IF ABS(Setpoint-Pos_value)<100 THEN
IF (Statusword.4)=1 THEN
State := 11;
END_IF

END_CASE
```

- Go to the left pane shown in Figure 3.10 (a) and select 'PlcTask'. Change the Cycle ticks as desired, **but not lower than 1 millisecond**.
- Furthermore, under the 'Real-Time' in the left pane shown in Figure 3.10 (b), select I/O Idle Task. Change the Cycle ticks as desired, **but not lower than 2 milliseconds**.

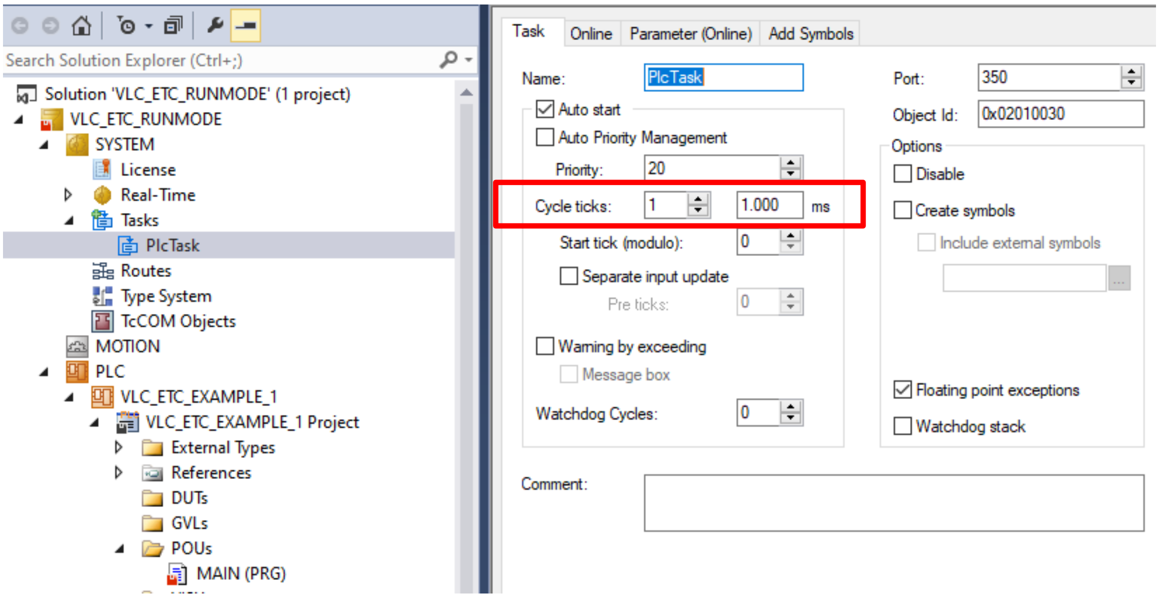


Figure 3.10 (a). Changing the scan time of the PLC program cycle.

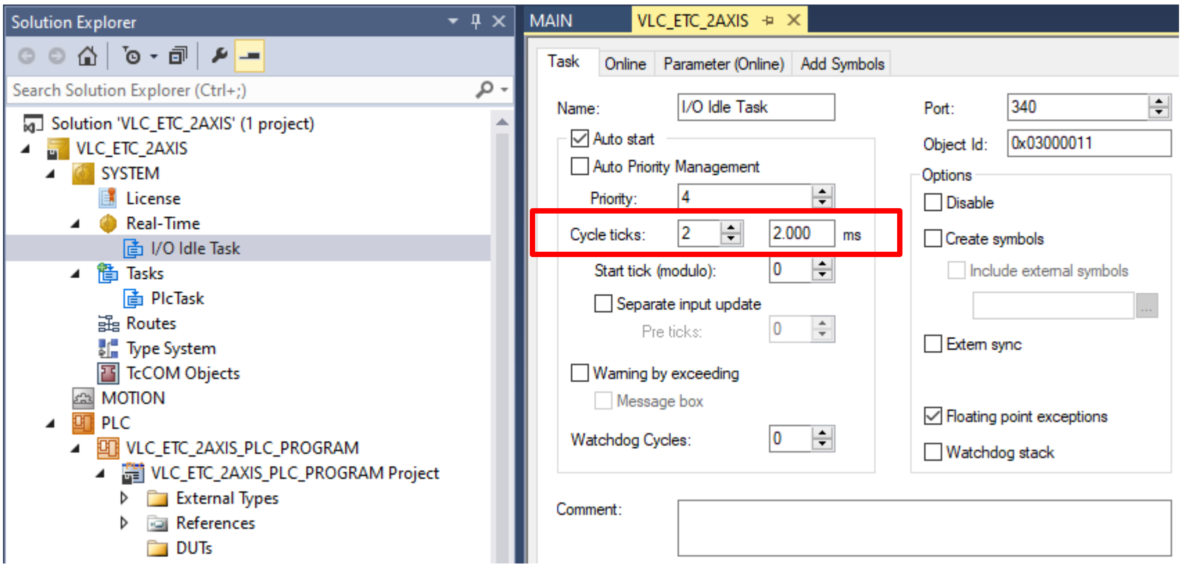


Figure 3.10 (b). Changing the scan time of the I/O update.

- Double-click on 'Real-Time' in the left pane of Fig. 3.11, then select the tab 'Priorities'. One can change the priority of the tasks in TwinCAT, or to optimize it by clicking on the button that's shown on the bottom of Figure 3.11.

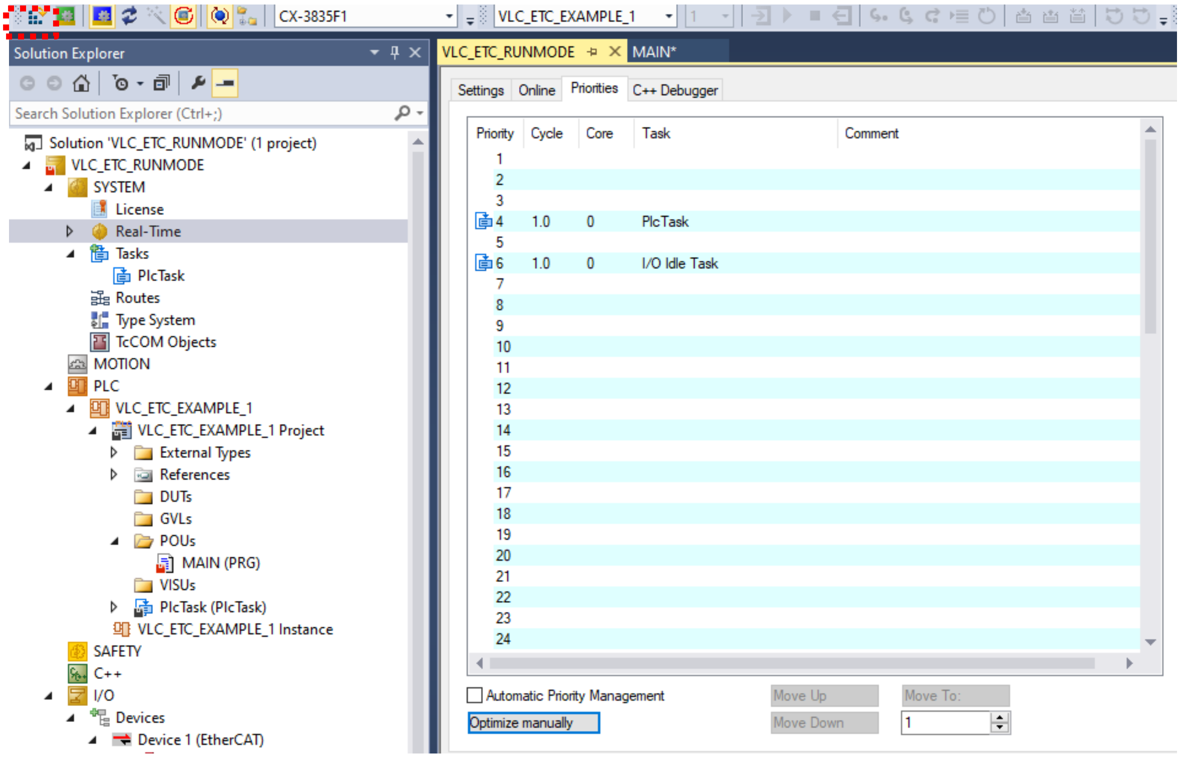


Figure 3.11. Changing task priorities.

- Click on 'Activate configuration' button on upper-right part of Figure 3.12. This will compile the PLC program and settings. PLC program syntax errors will be reported if they do exist. Otherwise, if the PLC program has never been compiled previously, a window in Figure 3.12 will appear, meaning that the variables defined in the program have to be linked to the cyclic servo objects of the VLC-ETC.
- After clicking OK on Figure 3.12, cancel the request to restart TwinCAT in Run Mode as shown in Figure 3.13.

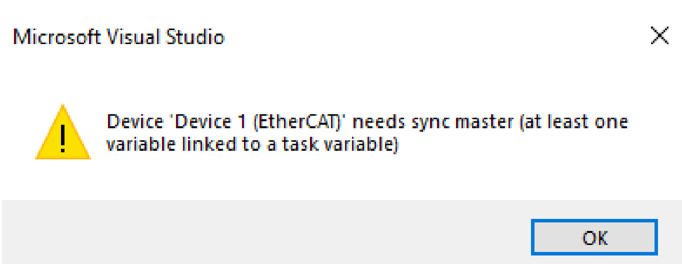


Figure 3.12. A warning that none of the variables have been linked.

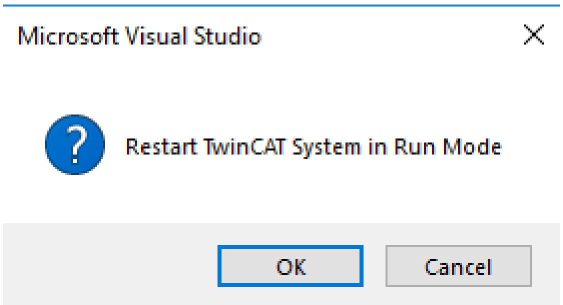


Figure 3.13.

- In the left-pane under the instances shown in Figure 3.14, PlcTaskInputs and Outputs appear now. These contain the variables that have been declared in the PLC program. Right-click on each of the variables and select 'Change link' and find the corresponding cyclic servo object of the VLC-ETC.
- Now, click again on the 'Activate configuration', this time, click OK to restart the TwinCAT in Run mode.
- Click on the login button in the rectangle in Figure 3.15 and select 'Yes' on the window that appears.
- To execute the PLC program, click on the 'Start' button in the rectangle in Figure 3.16. To stop the program execution, simply press the 'Stop' button next to the 'Start' button. Note that after stopping the program, controlword value will be set to 0 and therefore any macro execution in the VLC will be stopped and servo will be turned off.

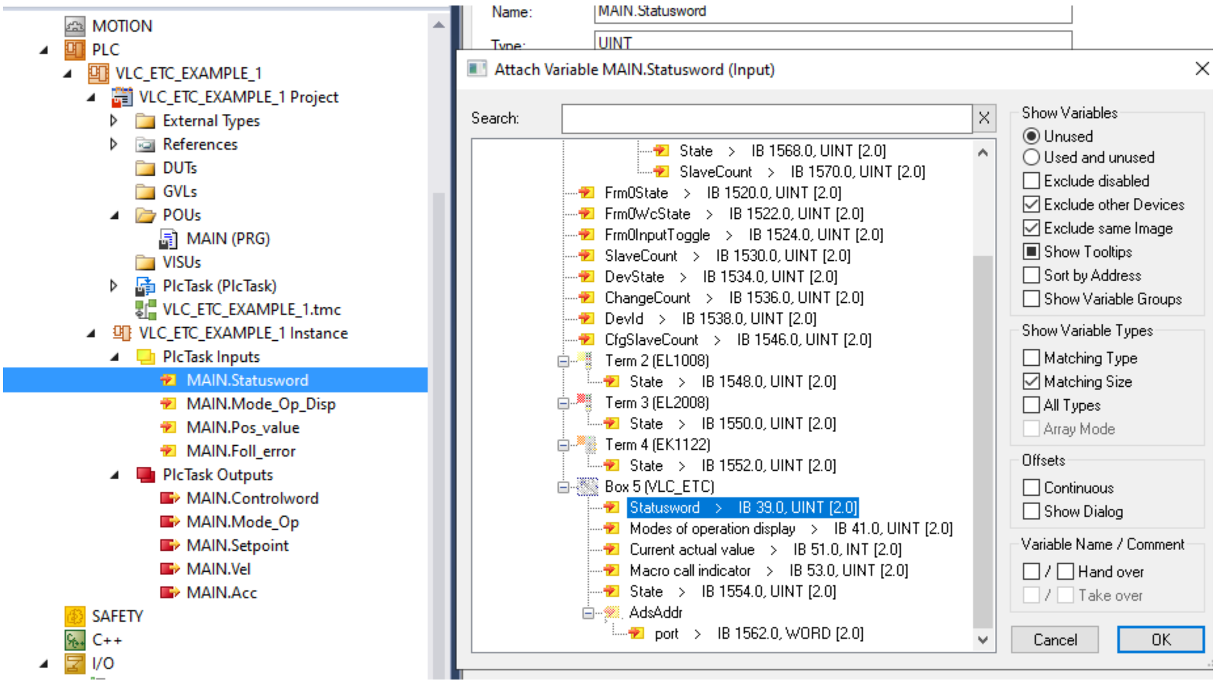


Figure 3.14. Linking the PLC input and output variables with the cyclic servo objects.

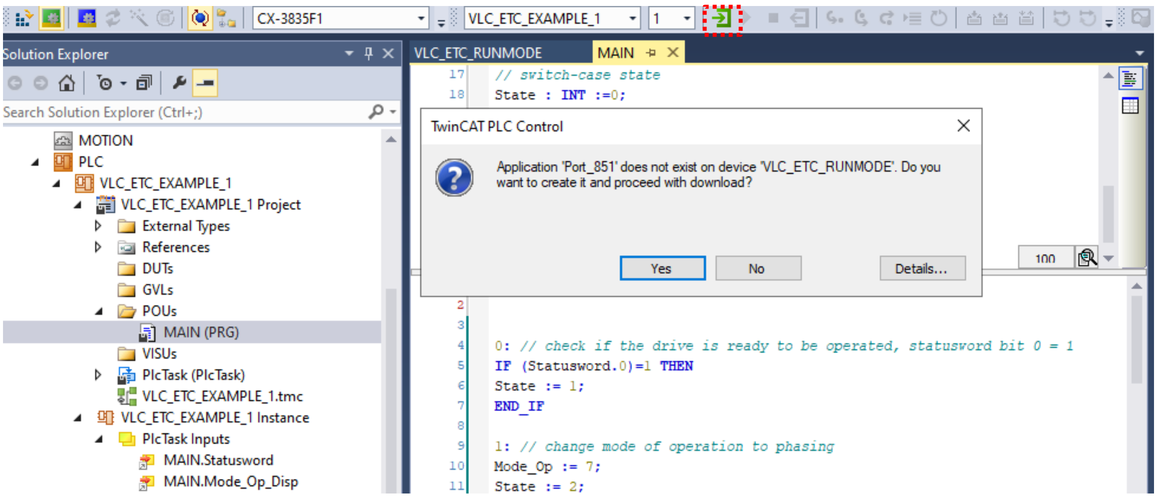


Figure 3.15. Login to the PLC program.

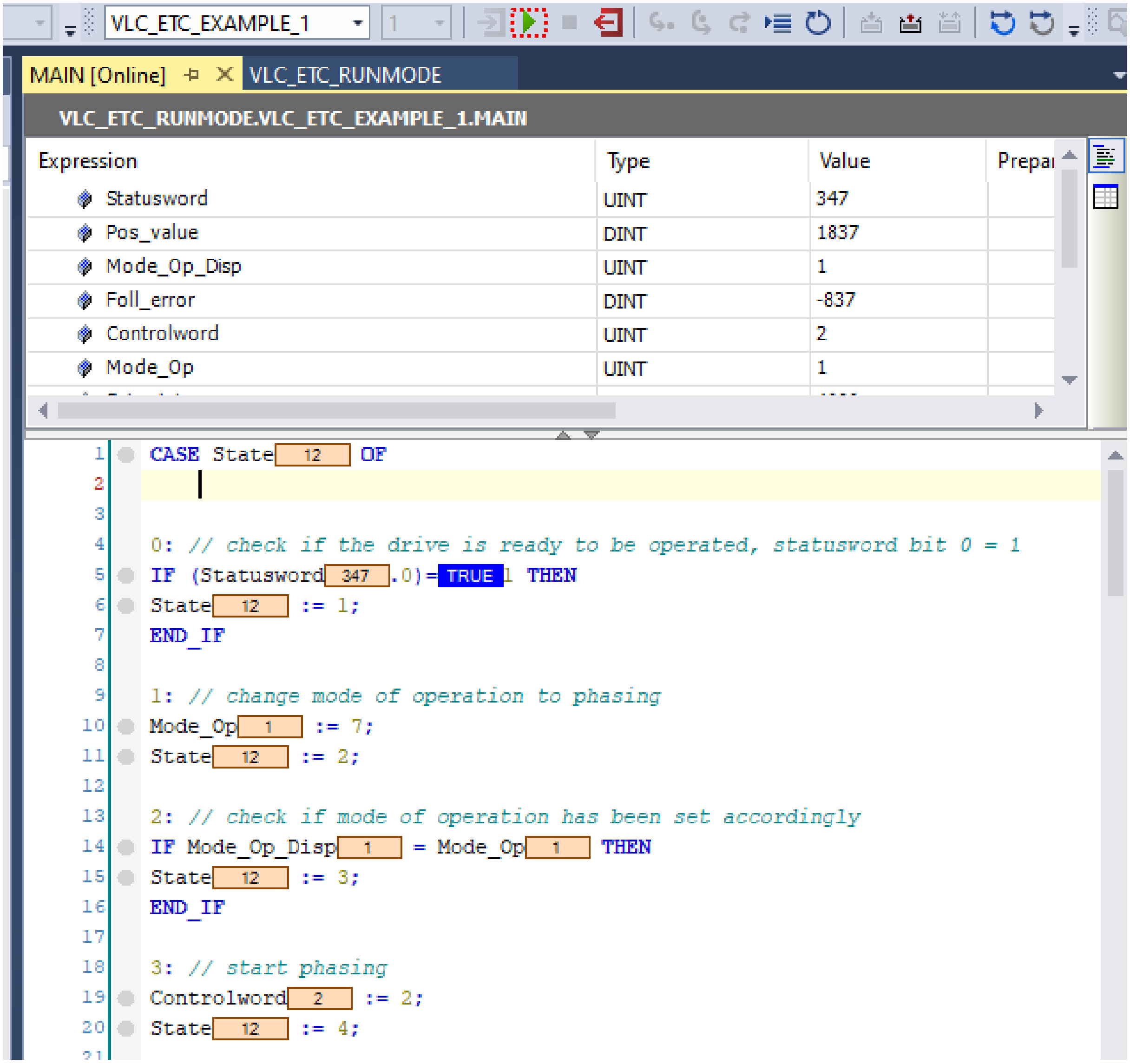
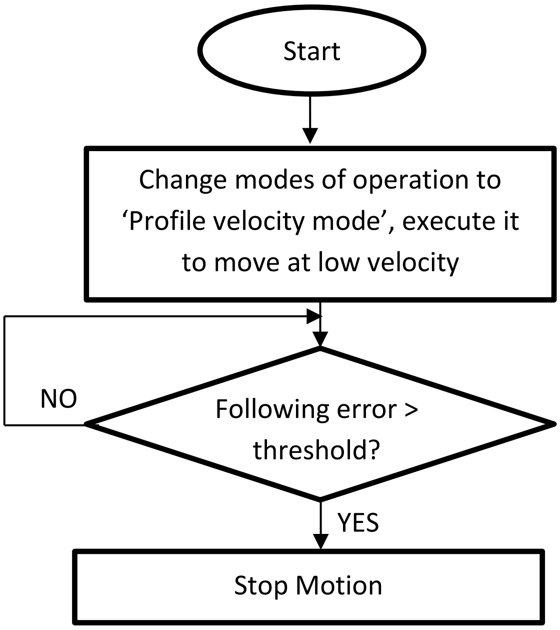


Figure 3.16. Executing the PLC program.

3.5 Example: programming a softland

Below is a flowchart of a basic softland routine



In VLC-ETC, there are two ways of executing the flowchart above:

- Algorithm implementation in a PLC program
  - Pros: a better overview of what is happening on the actuator at each step of the flowchart.
  - Cons: the detection of softland and the subsequent motion stopping occurs following the VLC-ETC cycle time, which has a lower limit of 2 msec. (as specified by the EtherCAT master).

- Calling a pre-programmed softland macro
  - Pros: faster softland detection as the following error monitoring happens internally in the VLC with an update rate in hundreds of microseconds range.
  - Cons: during the softland execution, the cyclic data update rate of the VLC-ETC becomes much slower. If the EtherCAT master cycle time is set to 2 msec., the cyclic data update rate during softland (or any other macro execution) would be around 12 msec.

3.5.1 Algorithm implementation in a PLC program

In a Structured Text (ST), similar to the previous example, the softland algorithm can be implemented using a CASE statement as follow (for the variable declaration, refer to example 3.4 of this manual):

```
15: // Velocity move
 Mode_Op := 3;
 State := 16;

16: // Check if modes of operation has been set accordingly
 IF (Mode_Op_Disp = Mode_Op) THEN
 State := 17;
 END_IF

17: // velocity move (softland) parameters
 Setpoint := 20000;
 Vel := Setpoint;
 Controlword := 2;// execute velocity move
 State := 18;


18: // check if motion acknowledgement bit has been set to 1
 IF (Statusword.3)=1 THEN
 Controlword := 1;
 State := 19;
 END_IF

19: // check if following errror exceeds threshold (softland is achieved)
 IF Foll_error > 400 THEN
 Controlword := 4;// stop
 State := 20;
 END_IF
```

3.5.2 Calling a pre-programmed softland macro

A macro sequence can be programmed in the VLC-ETC to execute the softland. An example is given in macro 250 of the system macros in Appendix A of this manual. The macro sequence utilizes general purpose registers (GPRs) accessible through the VLC-ETC cyclic servo objects, as softland parameters and status:

- GPR101: softland status (0: softland has not been achieved or is executing, 1: softland has been achieved)
- GPR103: softland velocity
- GPR104: softland following error threshold

Below is an example of executing softland through "macro execution" mode of operation, in Structured Text (for the variable declaration, refer to example 3.4 of this manual):

```
25: // macro execution mode
 Mode_Op := 10;
 State := 26;

26: // Check if modes of operation has been set accordingly
 IF (Mode_Op_Disp= Mode_Op) THEN
 State := 27;
 END_IF

27: // set softland parameters and execute macro call
 GPR103 := 20000; // softland velocity
 Vel := GPR103;
 GPR104 := 400; // error threshold
 Macro_call := 250;// softland macro
 Controlword := 2;// execute macro call
 State := 28;

28: // check if macro is being executed
 IF (Statusword.11)=1 AND GPR101 = 0  THEN
  Controlword := 1;
  State := 29;
 END_IF

29: // check if softland is achieved, then go to the next state
 IF GPR101 = 1 THEN
  State := 30;
 END_IF
```

3.6 Example: programming a 2-axis linear-rotary actuator

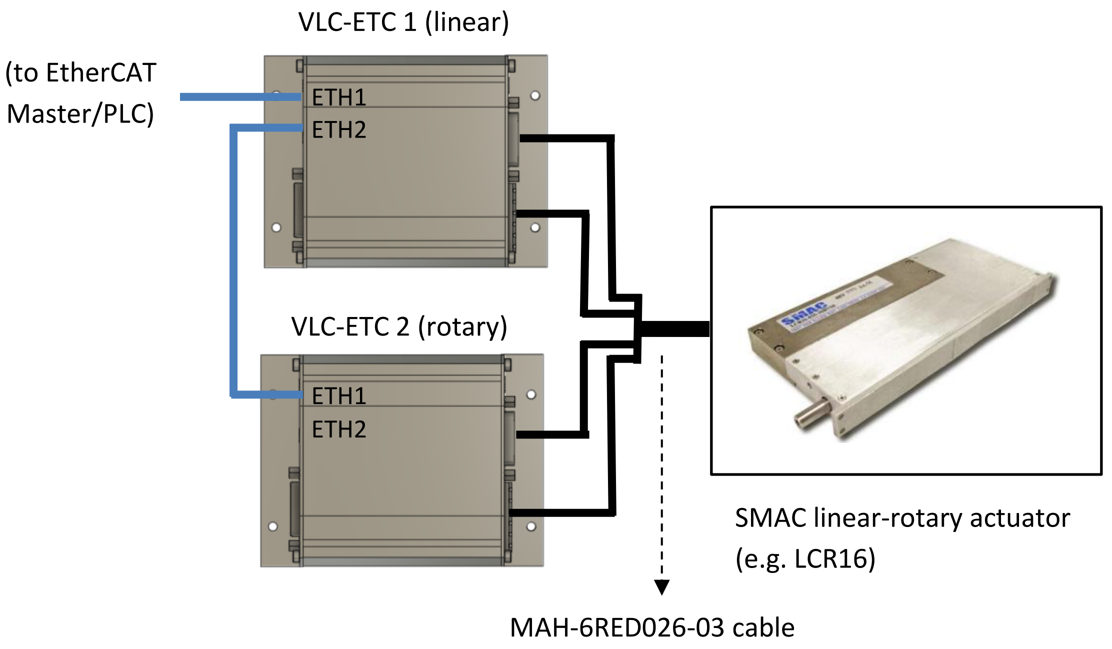


Figure 3.17. Example of a connection schematic for 2 VLC-ETCs (daisy-chained) and a 2-axis linear-rotary actuator.

Figure 3.17 depicts a case of 2 VLC-ETCs being used to drive a 2-axis linear-rotary actuator (LCR16-035) considered in this example. The motion task that the actuator has to perform is a sequence of phasing – homing – repetitive position move, for both axes. To prepare the program example in the context of TwinCAT, start a new project and follow the steps to 'scan for boxes' as explained in the previous example (section 3.4 of this manual), then follow these steps:

- Two VLC-ETCs will show up as shown in Figure 3.18.

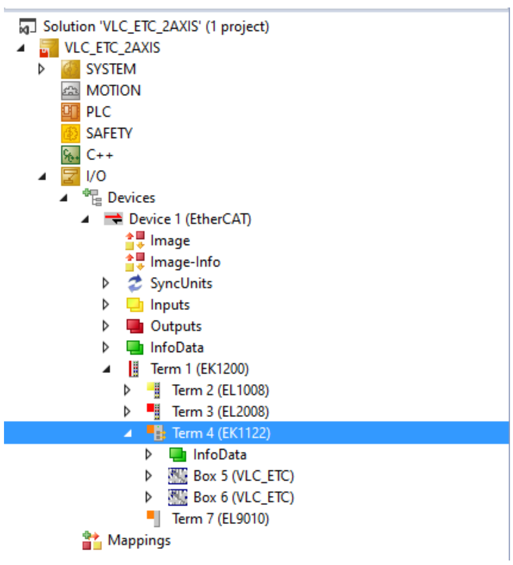


Figure 3.18. 2 VLC-ETCs detected in TwinCAT.

➢ Double-click on the VLC-ETC and select the "CoE-Online" tab as depicted in Figure 3.19. There is a table with various acyclic objects/configuration parameters such as controller PID settings.

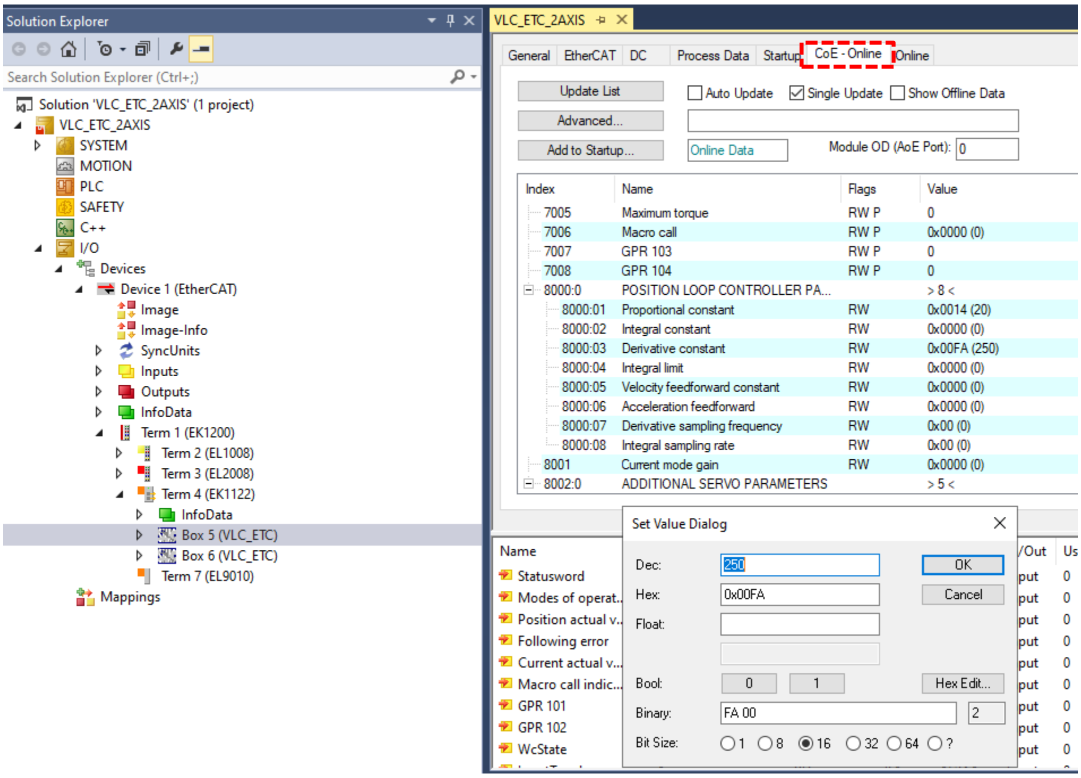


Figure 3.19. CoE-Online tab containing acyclic objects.

➢ In the CoE-Online tab, set the following parameters for each VLC-ETC

- Proportional constant
- Derivative constant
- Phase and sense settings
- Homing method
- Homing speed
- Homing acceleration
- Position error threshold
- Homing timeout
- Commutation electrical cycle
- Commutation voltage

Except for homing method, speed, acceleration, position error threshold and homing timeout, the above parameters could be determined from a test previously done actuator for each axis. Alternatively, the acyclic parameters in Table 3.6 can be used for an LCR16-035 actuator.

Table 3.6. Acyclic parameter values for an LCR16-035-15 actuator.

Parameter	**Box 5 (connected to linear actuator)**	**Box 6 (connected to rotary actuator)**
Proportional constant	20	3
Derivative constant	200	30
Phase and sense settings	1	1
Homing method	5	0
Homing speed	5000	500000
Homing acceleration	50000	500000
Position error threshold	500	200
Homing timeout	10000	10000
Commutation electrical cycle	3660	12000
Commutation voltage	10000	3000

- Finally, select the parameter 'Save all parameters' in the CoE-Online tab, change its value from 0 to 1.

- Right-click on the PLC in the left pane of TwinCAT environment and select 'Add new item'. Provide a name for a standard PLC project, as illustrated in Figure 3.20.

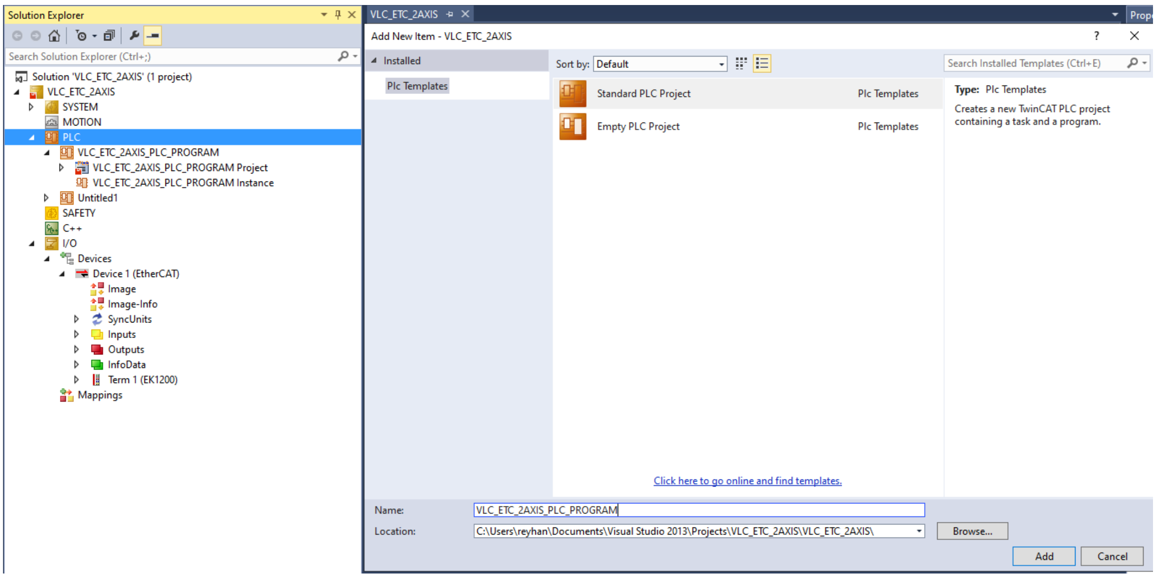


Figure 3.20.

➢ Go to the PLC program editor as shown in the left pane of Figure 3.21. The top rectangle is where the program variables are declared, while the bottom one is where the program logic will be located. In this example, the sequence of motions consists of: phasing – homing – repetitive position move.

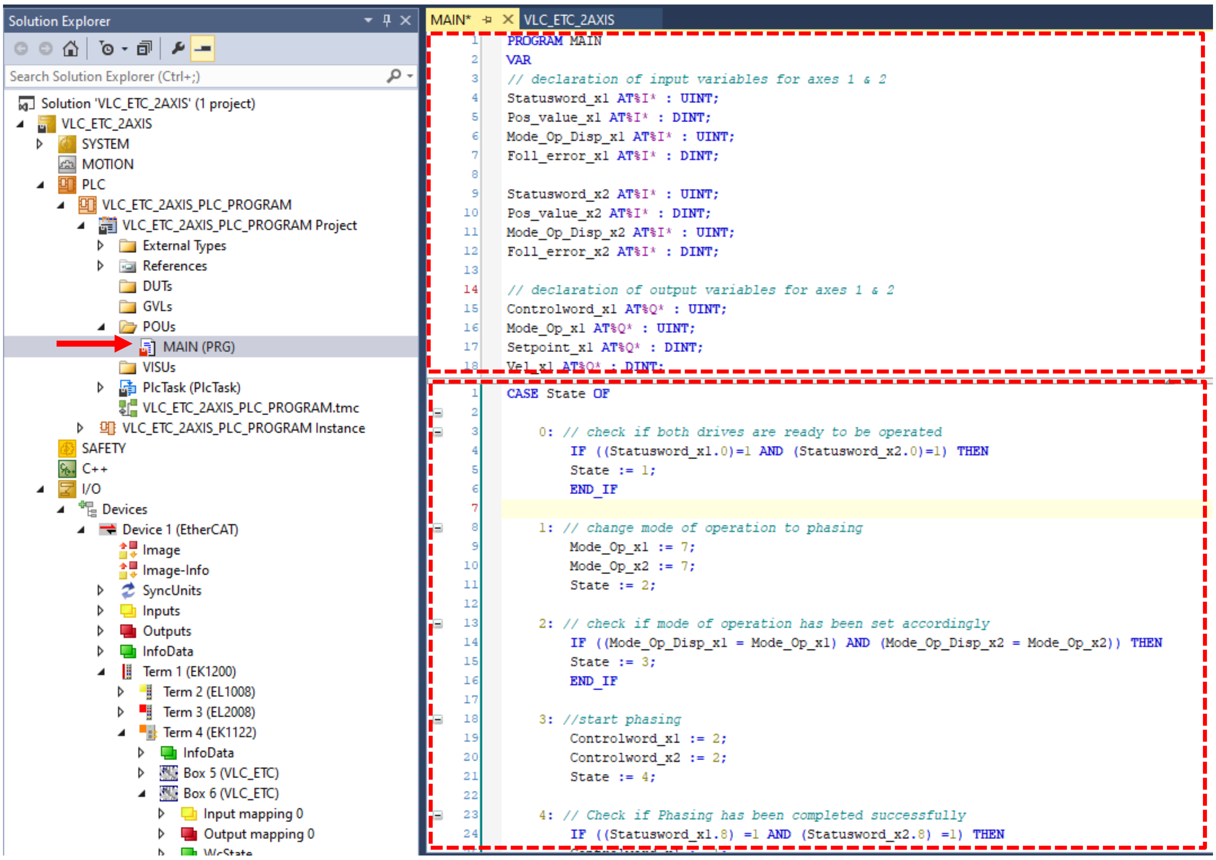


Figure 3.21.

➢ For the top part of the program editor, copy and paste the following

PROGRAM MAIN
VAR
// declaration of input variables for axes 1 & 2
Statusword_x1 AT%I* : UINT;
Pos_value_x1 AT%I* : DINT;
Mode_Op_Disp_x1 AT%I* : UINT;
Foll_error_x1 AT%I* : DINT;

Statusword_x2 AT%I* : UINT;
Pos_value_x2 AT%I* : DINT;
Mode_Op_Disp_x2 AT%I* : UINT;
Foll_error_x2 AT%I* : DINT;

```
// declaration of output variables for axes 1 & 2
Controlword_x1 AT%Q* : UINT;
Mode_Op_x1 AT%Q* : UINT;
Setpoint_x1 AT%Q* : DINT;
Vel_x1 AT%Q* : DINT;
Acc_x1 AT%Q* : DINT;

Controlword_x2 AT%Q* : UINT;
Mode_Op_x2 AT%Q* : UINT;
Setpoint_x2 AT%Q* : DINT;
Vel_x2 AT%Q* : DINT;
Acc_x2 AT%Q* : DINT;

// switch-case state
State : INT :=0;
pos_state : INT;

// phasing & homing error indicator
Phasing_Error_x1 : BOOL := FALSE;
Homing_Error_x1 : BOOL := FALSE;

Phasing_Error_x2 : BOOL := FALSE;
Homing_Error_x2 : BOOL := FALSE;

// Timer
 Timer: TON;
END_VAR
```

- For the bottom part of the program editor, copy and paste the following

```
CASE State OF

 0: // check if both drives are ready to be operated
  IF ((Statusword_x1.0)=1 AND (Statusword_x2.0)=1) THEN
  State := 1;
  END_IF

 1: // change mode of operation to phasing
  Mode_Op_x1 := 7;
  Mode_Op_x2 := 7;
  State := 2;
```

```
2: // check if mode of operation has been set accordingly
 IF ((Mode_Op_Disp_x1 = Mode_Op_x1) AND (Mode_Op_Disp_x2 = Mode_Op_x2)) THEN
 State := 3;
 END_IF

3: //start phasing
 Controlword_x1 := 2;
 Controlword_x2 := 2;
 State := 4;

4: // Check if Phasing has been completed successfully
 IF ((Statusword_x1.8) =1 AND (Statusword_x2.8) =1) THEN
 Controlword_x1 := 1;
 Controlword_x2 := 1;
 Phasing_Error_x1 := FALSE;
 Phasing_Error_x2 := FALSE;
 State := 5;
 END_IF

 IF (Statusword_x1.9) = 1 THEN
  Phasing_Error_x1 := TRUE;
 END_IF

 IF (Statusword_x2.9) = 1 THEN
  Phasing_Error_x2 := TRUE;
 END_IF

5: // Homing
 Mode_Op_x1 := 6;
 Mode_Op_x2 := 6;
 State := 6;

6: // Check if modes of operation has been set
 IF ((Mode_Op_Disp_x1 = Mode_Op_x1) AND (Mode_Op_Disp_x2 = Mode_Op_x2)) THEN
 State := 7;
 END_IF

7: // Execute homing
 Controlword_x1 := 2;
 Controlword_x2 := 2;
 State := 8;
```

```
8: // check if homing has been done successfully
 IF ((Statusword_x1.6) =1 AND (Statusword_x2.6)=1) THEN
 Controlword_x1 := 1;
 Controlword_x2 := 1;
 Homing_Error_x1 := FALSE;
 Homing_Error_x2 := FALSE;
 State := 9;
 END_IF

 IF (Statusword_x1.7) = 1 THEN
  Homing_Error_x1 := TRUE;
 END_IF

 IF (Statusword_x2.7) = 1 THEN
  Homing_Error_x1 := TRUE;
 END_IF

9: // change mode of operation to position move (absolute)
 Mode_Op_x1 := 1;
 Mode_Op_x2 := 1;
 State := 10;

10: // check if mode of operation has been changed
 IF ((Mode_Op_Disp_x1 = Mode_Op_x1) AND (Mode_Op_Disp_x2 = Mode_Op_x2)) THEN
 State := 11;
 END_IF

11:
 Vel_x1 := 100000;
 Acc_x1 := 100000;
 Setpoint_x1 := 4000;
 Controlword_x1 := 2;

 Vel_x2 := 10000000;
 Acc_x2 := 10000000;
 Setpoint_x2 := 480000;
 Controlword_x2 := 2;
 pos_state := 11;
 State := 20;

12:
 Vel_x1 := 100000;
 Acc_x1 := 100000;
 Setpoint_x1 := 1000;
 Controlword_x1 := 2;
```

```
  Vel_x2 := 10000000;
  Acc_x2 := 10000000;
  Setpoint_x2 := 0;
  Controlword_x2 := 2;
  pos_state := 12;
  State := 20;

 20: // check if motion acknowledgement bit has been set to 1
  IF ((Statusword_x1.3)=1 AND (Statusword_x2.3)=1) THEN
  Controlword_x1 := 1;
  Controlword_x2 := 1;
  State := 21;
  END_IF

 21: // wait for 1 second then go back to State 7
  Timer.IN := TRUE;
  Timer.PT := T#1.0S;
  IF Timer.Q THEN
   Timer.IN := FALSE;
   IF pos_state = 11 THEN
   State := 12;
   END_IF
   IF pos_state = 12 THEN
   State := 11;
   END_IF
  END_IF
  Timer();

END_CASE
```

- Click on the 'Activate Configuration' button on the upper left rectangle of Figure 3.22. This will compile the PLC program and settings. PLC program syntax errors will be reported if they do exist. Otherwise, if the PLC program has never been compiled previously, a window depicted in Figure 3.23 appears, meaning that the variables defined in the program have to be linked to the cyclic servo objects of the VLC-ETC.

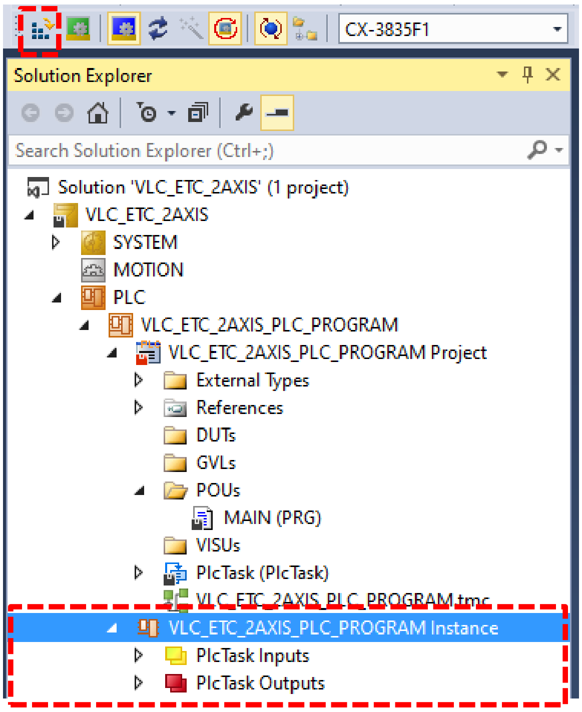


Figure 3.22.

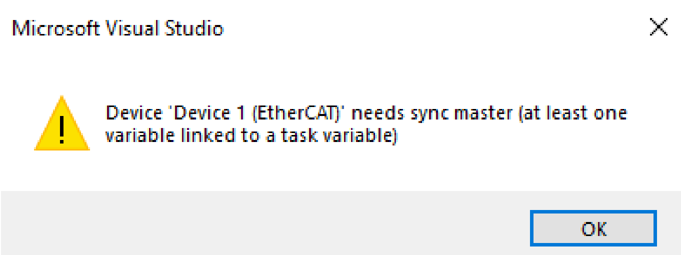


Figure 3.23.

- After clicking OK in the previous window, cancel the request to restart TwinCAT in Run Mode shown in Figure 3.24.

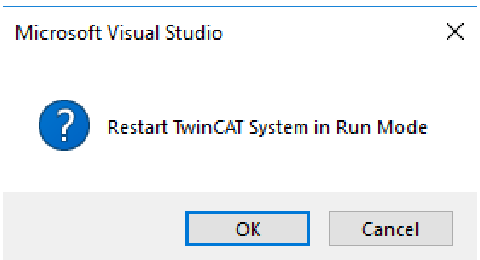


Figure 3.24.

➢ As shown in Figure 3.25, expand the ‘PlcTask Inputs’ and ‘PlcTask Outputs’, under which there is a list of input and output variables that are defined in the PLC program editor. Right-click on each of the variables and link it to the corresponding variable with a similar name belonging to the VLC-ETC. Variables with ‘_x1’ suffix are to be linked with the VLC-ETC connected to the linear axis, while ‘_x2’ corresponds to the rotary axis.

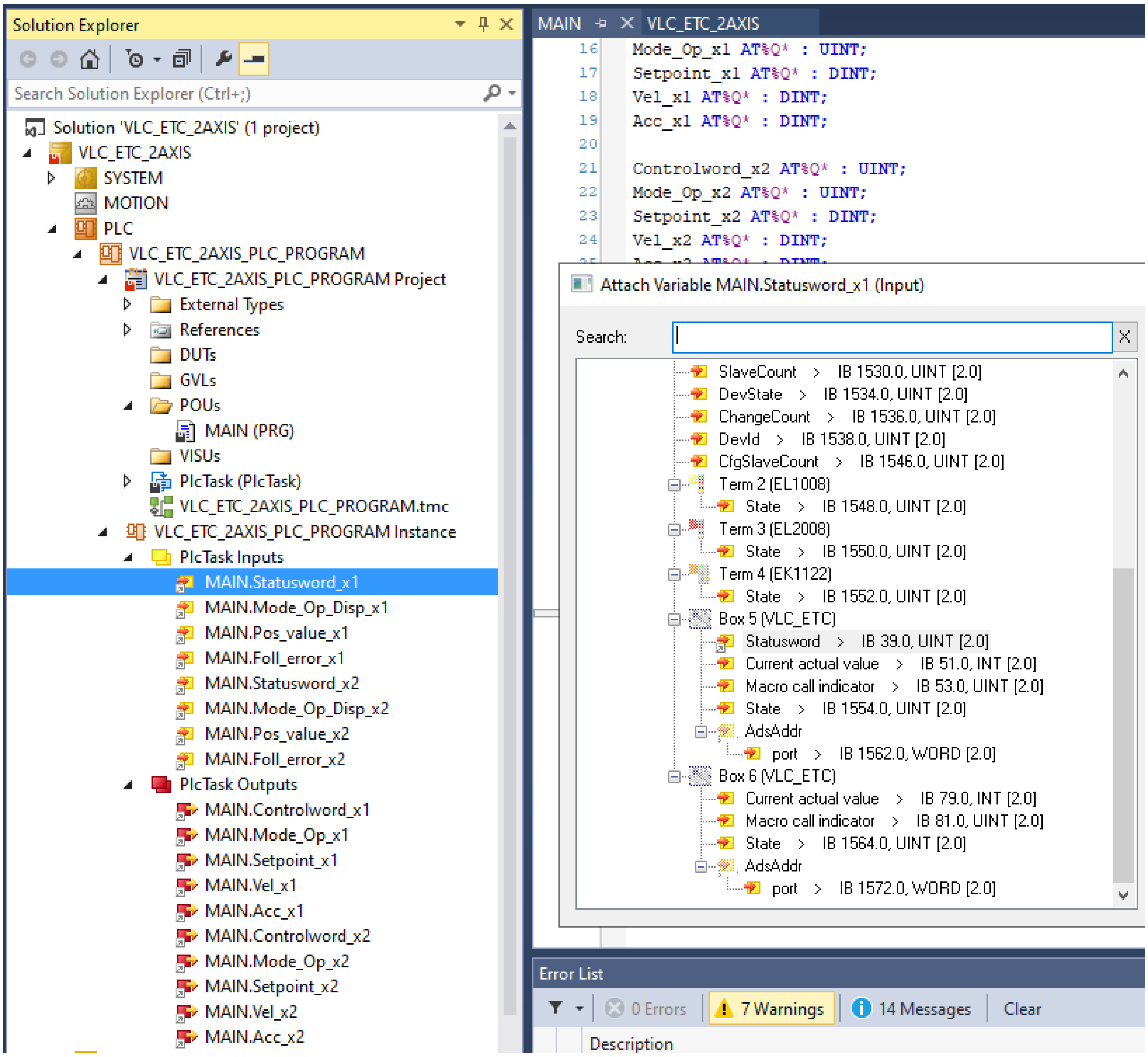


Figure 3.25.

➢ On the left-pane of Figure 3.26, select 'Real-Time' and go to the 'Priorities' tab and click on the 'Optimize manually' button highlighted in the below screen.

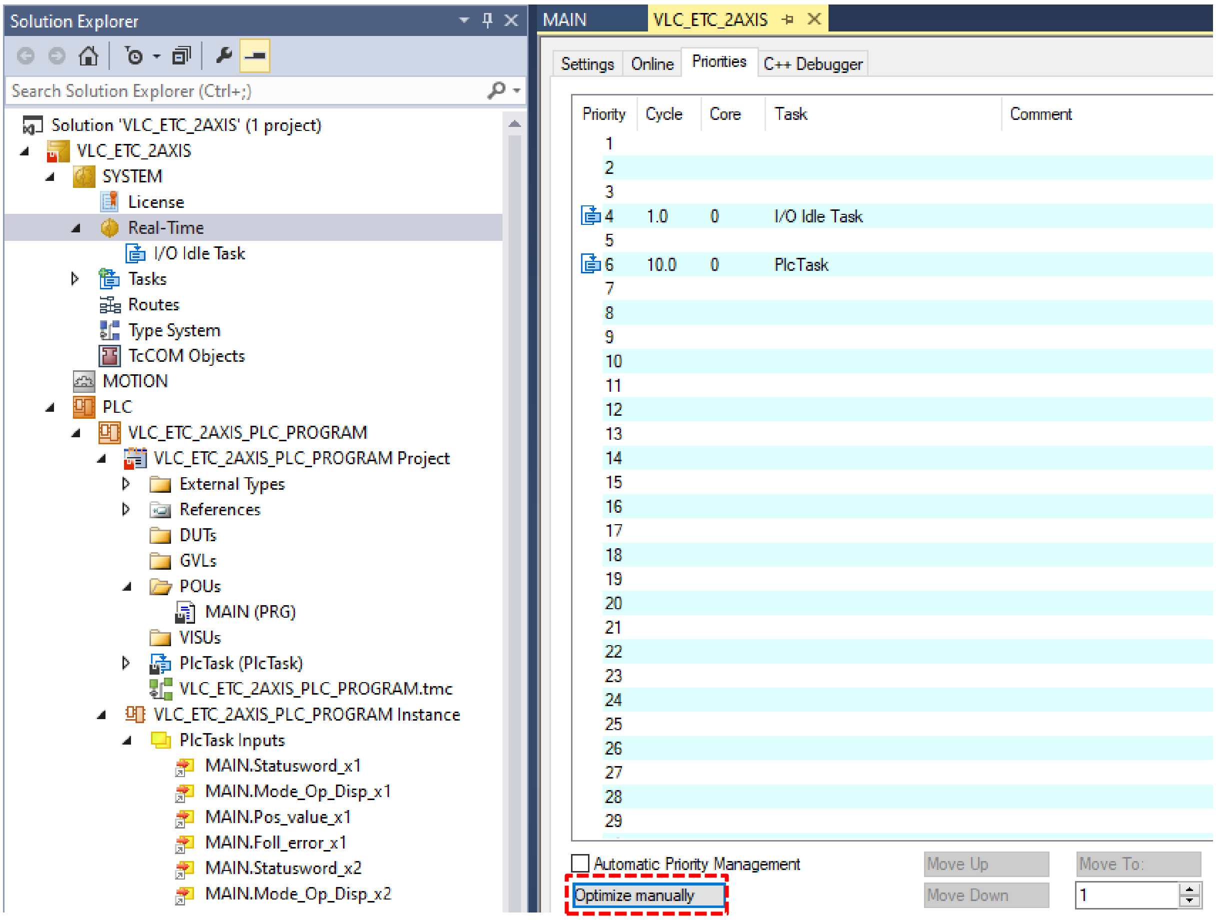


Figure 3.26.

➢ Select 'I/O Idle Task' as can be seen in Figure 3.27, change the cycle ticks to 2.

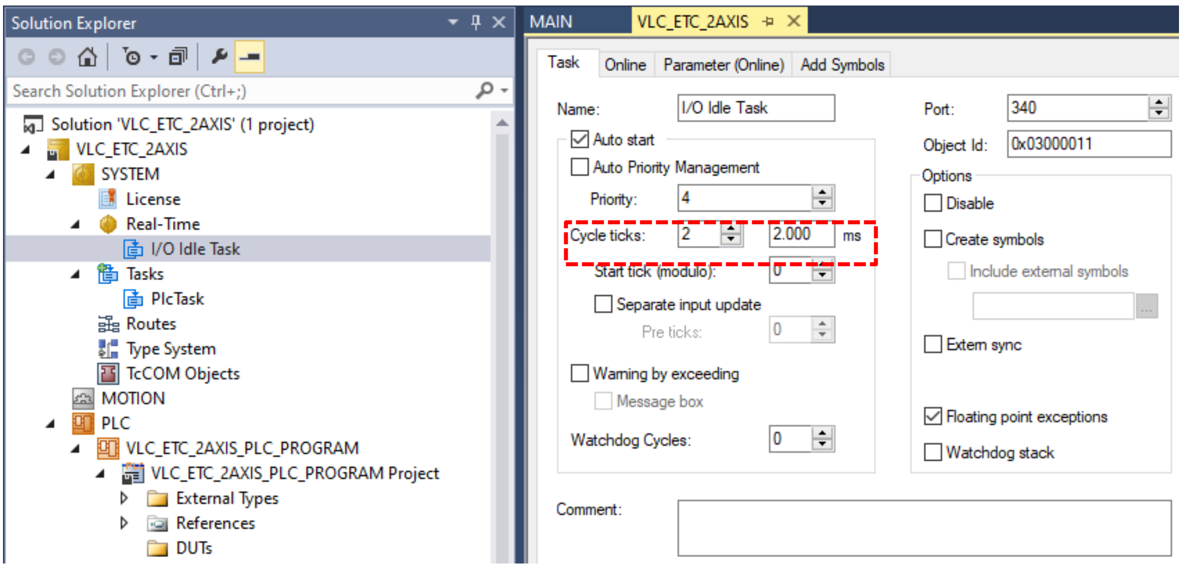


Figure 3.27.

➢ Select ‘Plc Task’ as shown in Figure 3.28, change the cycle ticks to 1.

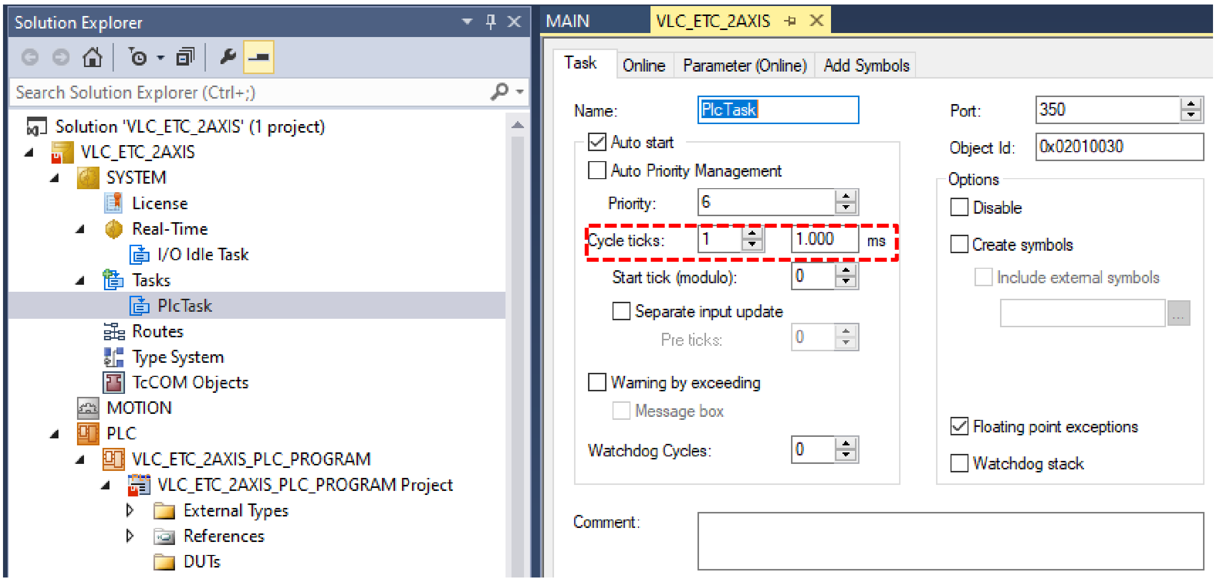


Figure 3.28.

➢ Click on the ‘Activate configuration’ icon highlighted in Figure 3.29, click OK, and accept the request to restart TwinCAT in Run Mode

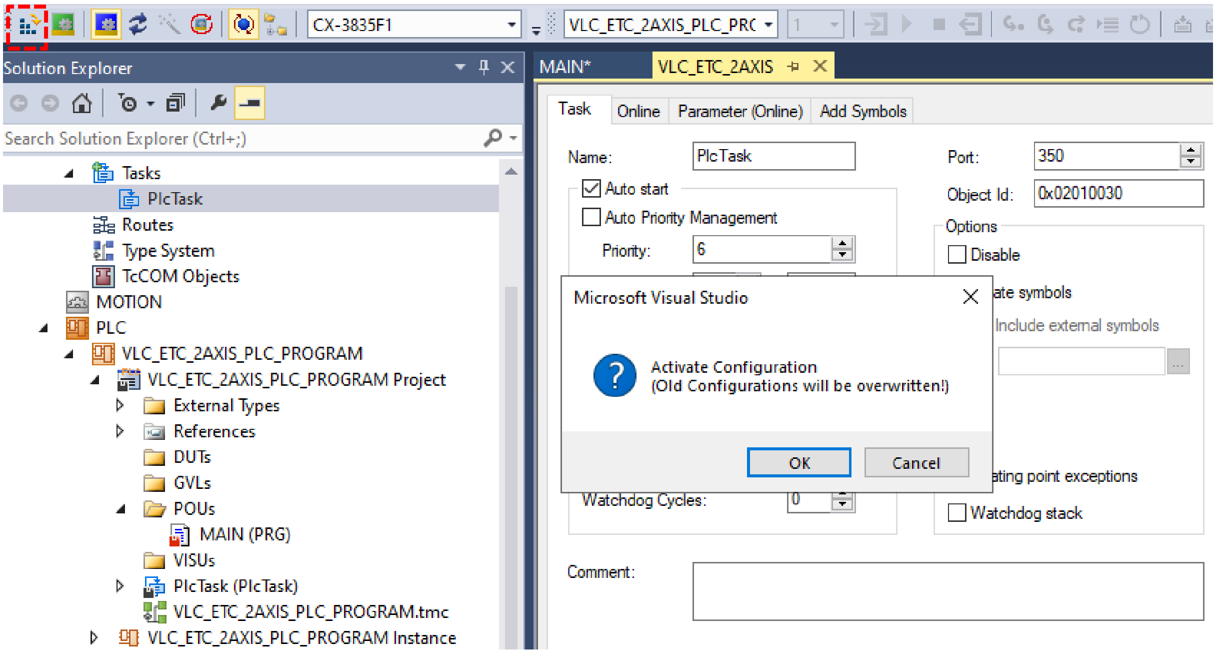


Figure 3.29.

➢ Click on the 'Login' icon highlighted in Figure 3.30, select Yes when there is a request to proceed with download

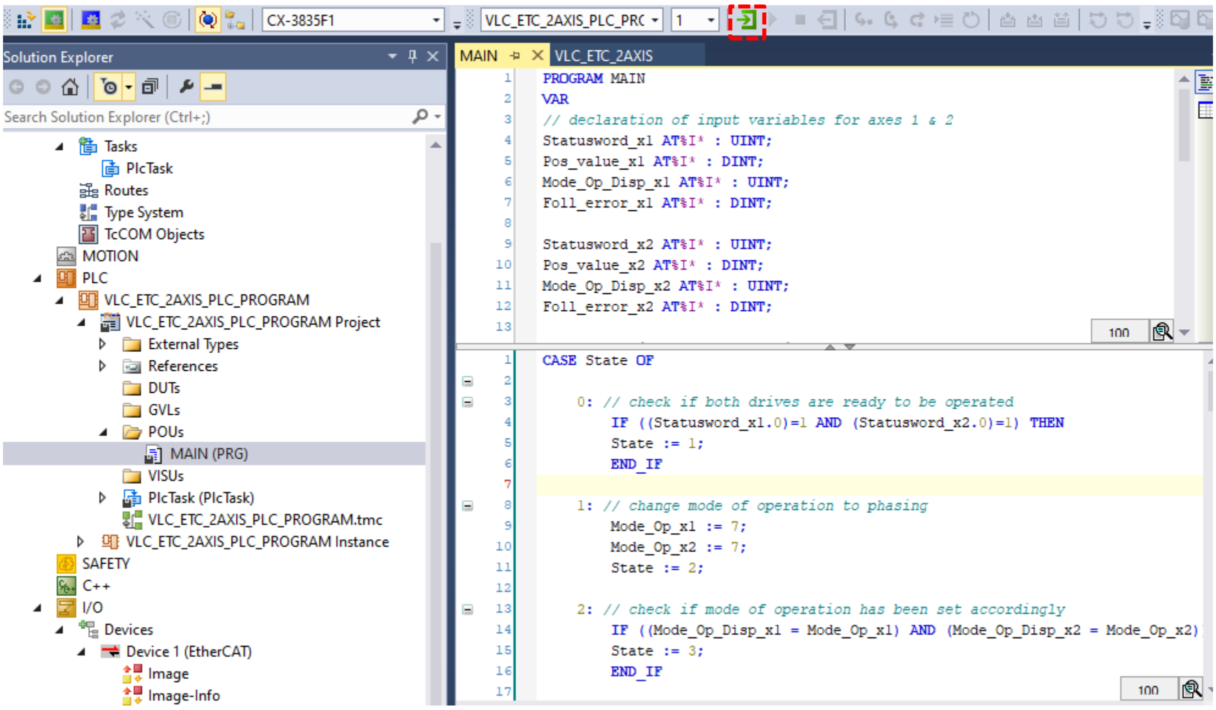


Figure 3.30.

➢ Lastly, click on the button highlighted in Figure 3.31 to execute the program.

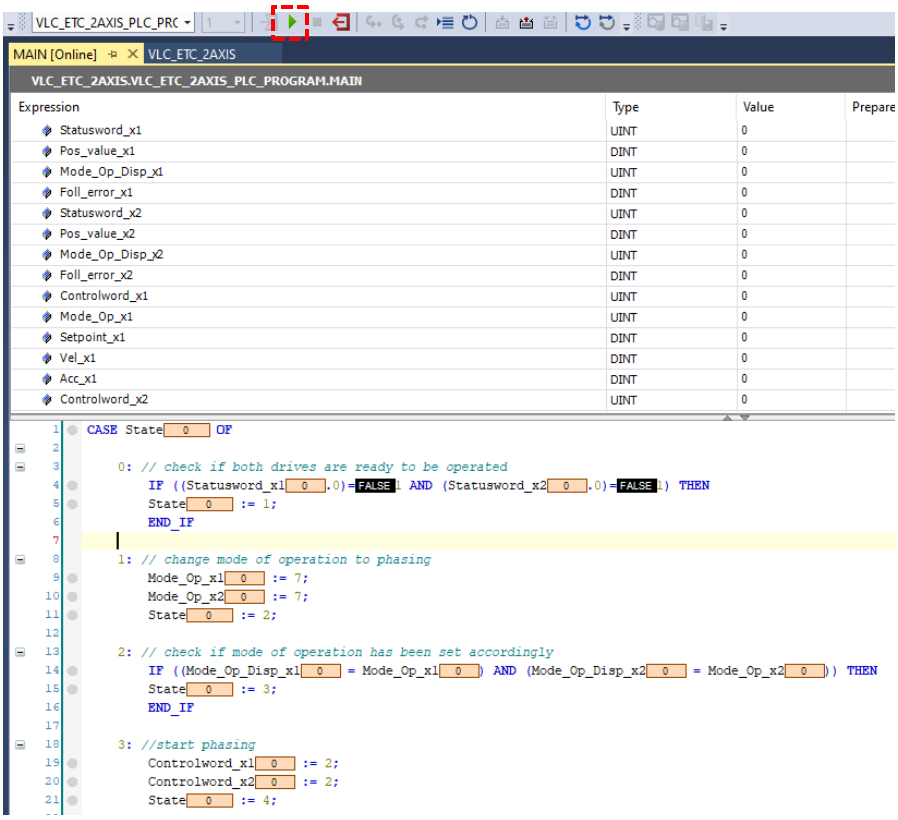


Figure 3.31.

- In case the previous program does not run: look at the variable ‘State’, which indicates the part where the program gets stuck. This could indicate that the statusword bit corresponding to phasing or homing has not been set to 1, meaning that there is a failure in phasing or homing. Make sure that the parameters are set properly and tested before. Refer to section 3.3 of this manual for a guide on testing motion manually through TwinCAT config mode (no PLC is required).

Moving Coil Actuators

A Appendix A: VLC System Macros

```
; Initialization
MD0,EF,BR460800,AL0,AR101,AR102,AR214,AR224,MJ151

; acyclic object parameters
MD151,SG0,SI0,SD0,IL0,FV0,FA0,FR0,RI0,SC0,SS2,PH0
MD152,DB0,OO0,SE16383,AL0,AR220,AL5000,AR215,AL5000,AR216,AL0,AR221
MD153,AL200,AR222,AL5000,AR225,AL0,SP0,AL5000,AR200,AL0,AR201
MD154,DA0,AL2047,WW610,AL2147483647,WL612,EP

; initial acyclic objects check
MD155,RW516,TR,RW518,TR,RW520,TR,RW522,TR,RW526,TR,RW536,TR,RB550,TR,RB552,
TR,RW524,TR,RB1822,TR,RB558,TR,RW560,TR,EP
MD156,RW528,TR,RW542,TR,TR220,TR215,TR216,TR221,TR222,TR225,RW604,TR,TR200,
TR201,RL592,TR,EP
MD157,TR11,TR12,TR13,TR14,TR15,TR16,TR17,TR18,TR19,TR20,EP
MD158,TR21,TR22,TR23,TR24,TR25,RW610,TR,RL612,TR,EP

; periodic cyclic objects read
MD159,TS,TP,TF,RW548,TR,TR101,TR102,BI0,TR,RC

; Phasing system macros (R200: SQ, R201: EC, R214: phasing status)
MD200,AL0,AR214,MC245
MD201,MF,EC0,AL32767,AR202,AL16384,AR204,AM@201,IG0,AD65536,MJ202,RA201,AD6
5536,AM@204
MD202,AR205,AM9,AD10,AR206,RA205,AM11,AD10,AR207
MD203,SP0,QM0,MN,SQ@200,WA100,MC211,AL1,AR229,MC208
MD204,AL0,AA@204,AR209,SP@209,WA100,MC211,AL3,AR229,MC208,AL1,AR229,MC209
MD205,AL65535,AS@204,AR209,SP@209,WA100,MC211,AL5,AR229,MC208,AL4,AR229,MC2
09
MD206,SP@202,WA100,MC211,AL1,AR229,MC208,AL1,AR229,MC210,AL3,AR229,MC208,AL
1,AR229,MC209
MD207,AL4,AR229,MC210,AL5,AR229,MC208,AL4,AR229,MC209,MJ215
MD208,RL494,JR@229,AR203,JR4,AR208,JR2,AR228,RC
MD209,JR@229,RA208,AS@203,JR3,RA203,AS@228,IG@206,IB@207,MJ212,RC,RC
MD210,JR@229,RA202,AA@204,JR3,RA202,AS@204,AR209,SP@209,WA100,MC211,RC
MD211,RL494,AR211,WA10,RL494,AS@211,IE0,NO,RC,RP
MD212,RW556,IB0,AA65536,NO,IU@202,IU@209,RW604,NO,IB0,AA65535,AR210,AR210
MD213,RA201,AD4,AR212,RA210,AM@201,AD65535,AA@212,AR213,DA@213,EC@201,SQ0
MD214,AL1,AR214,UM1,MF0,EP
MD215,AL2,AR214,SQ0,MF0

; Position move system macros (R217: target position)
MD217,PM,MN,MA@217,GO
MD219,PM,MN,MR@217,GO

; Velocity move system macros (R217: target velocity)
MD221,RA217,IG0,DI0,MJ222,DI1,AM-1,AR217
MD222,SV@217,VM,MN,GO
```

```
; Torque move system macros (R200: SQ value)
MD224,QM0,MN,SQ@200
MD226,QM1,MN,SQ@200

; Homing system macros (R215: velocity, R216: acceleration, R220: homing method <0-current position, 1-negative mech limit, 2-positive mech limit, 3-negative index, 4- positive index, 5-negative mech limit and index, 6-positive mech limit and index>, R221: home offset, R222: error,  R224: homing status, R225: timeout)
MD229,AL0,AR224,SV@215,SA@216,RL1830,AR226,RA222,AM-1,AR223,RA220
MD230,IE0,DH@221,EP,IE1,DI1,MJ231,IE2,DI0,MJ232,IE3,DI1,MJ233,IE4,DI0,MJ233
,IE5,DI1,MJ234,IE6,DI0,MJ235
MD231,MC236,MC237,MJ239
MD232,MC236,MC238,MJ239
MD233,MC236,FI,MJ240
MD234,MC236,MC237,ST,WA50,DI0,MC236,FI,MJ240
MD235,MC236,MC238,ST,WA50,DI1,MC236,FI,MJ240
MD236,VM,MN,GO,RC
MD237,RW538,IB@223,NO,RC,RL1830,AS@226,IG@225,NO,MJ241,RP
MD238,RW538,IG@222,NO,RC,RL1830,AS@226,IG@225,NO,MJ241,RP
MD239,ST,DH@221,WS,AL1,AR224,EP
MD240,RL448,AN1024,IE0,MJ239,NO,RL1830,AS@226,IG@225,NO,MJ241,RP
MD241,MF,AL2,AR224,EP

; Automatic current sensing offset adjustment. These are called when phasing is performed.
MD245,AL2048,WW606,WW608,AL0,AR400,AR401
MD246,GA1,AA@400,AR400,GA2,AA@401,AR401,WA1,RP999
MD247,RA400,AM-1,AD1000,AA2048,WW606
MD248,RA401,AM-1,AD1000,AA2048,WW608,RC

; Softland routine (R101: softland status <1- softland reached>, R103: velocity, R104: position error threshold)
MD250,AL0,AR101,RA103,IG0,DI0,MJ251,MJ253
MD251,SV@103,VM,MN,GO,WA50
MD252,RW538,IG@104,MJ255,NO,RP
MD253,DI1,AM-1,AR103,RA104,AM-1,AR104,SV@103,VM,MN,GO,WA50
MD254,RW538,IB@104,MJ255,NO,RP
MD255,ST,AL1,AR101
```