Tolomatic
EXCELLENCE IN MOTION

Technical Note

TwinCAT® 3 Integration Guide

ACSI Integrated Servo Motor/Drive/Controller

EtherCAT®

3600-4202_00

Contents

Device Information
Vendor ID: 0x00000986
Vendor Name: Tolomatic, Inc.
Product Code: 0x2362
Type: ACSI Drive & Controller
Name: ACSI Drive & Controller Interface

For use with 3604-3324_ACSI_ESI definitions file.

For use with 3604-3325 ACSI TwinCAT Library.

Introduction

This integration guide provides step by step instructions on how to get an ACS Integrated Motor up and running in a TwinCAT 3 Environment. This guide assumes that the user is familiar with commissioning devices in the TwinCAT 3 environment, and is not intended to provide all necessary steps for commissioning a PLC.

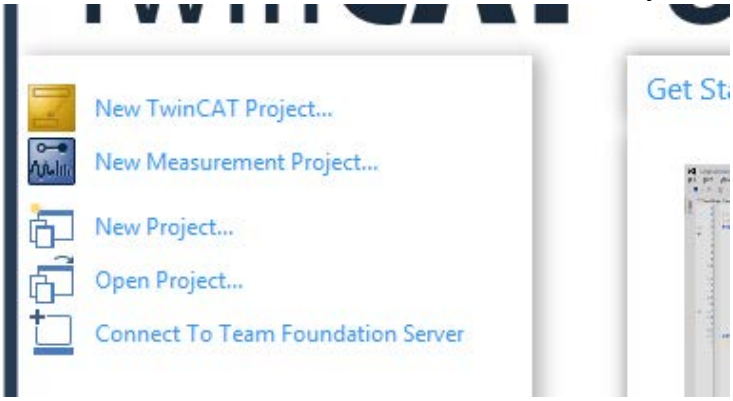
Additional Information

Additional information for the ACSI EtherCAT implementation can be found in the ACSI Hardware User's Guide, the EtherCAT User's Guide, and the Tolomatic Motion Interface User's Guide.

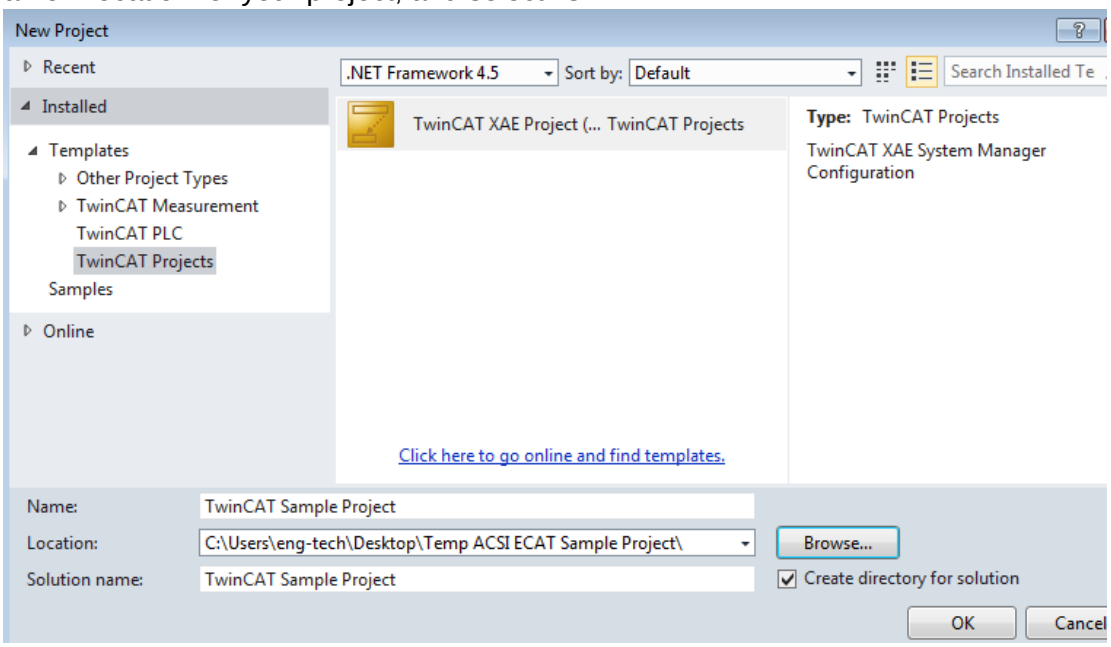
Initial Setup

TwinCAT 3 installs with a Visual Studio 2013 Shell.

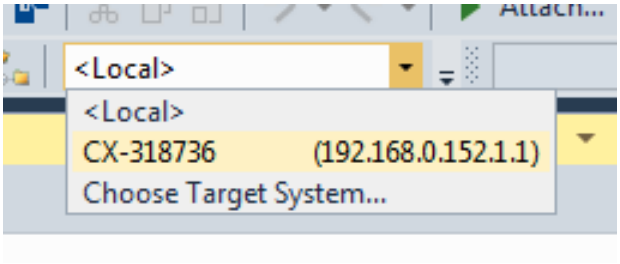
1. Launch Visual Studio 2013, and select "New TwinCAT Project…"



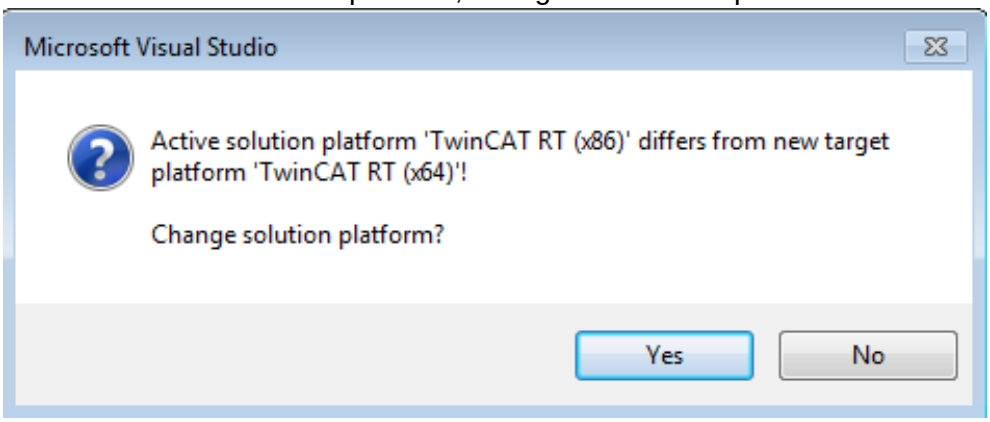
2. Create a new location for your project, and select “OK”



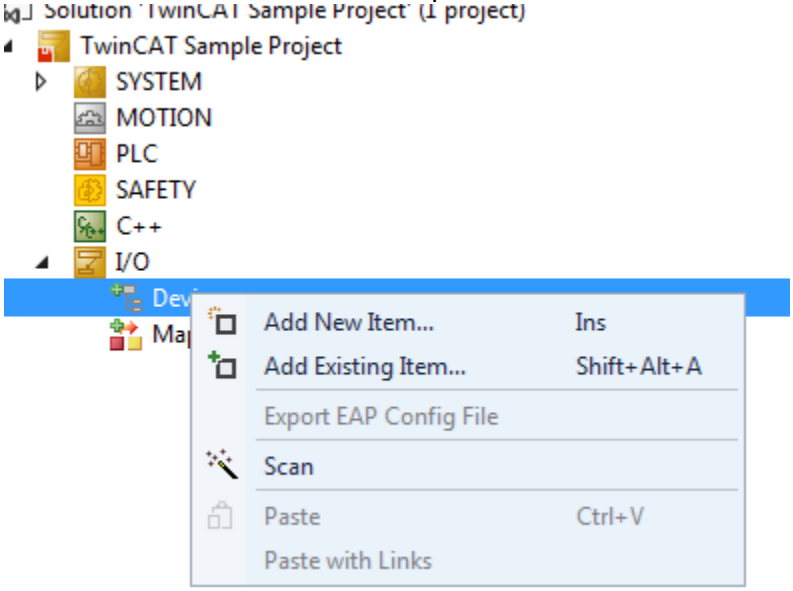
3. Select Target System (PLC) for your project. This can be “Local” for a simulated TWinCAT PLC, or a defined route. More information about setting up routes can be found in the Beckhoff TwinCAT documentation.



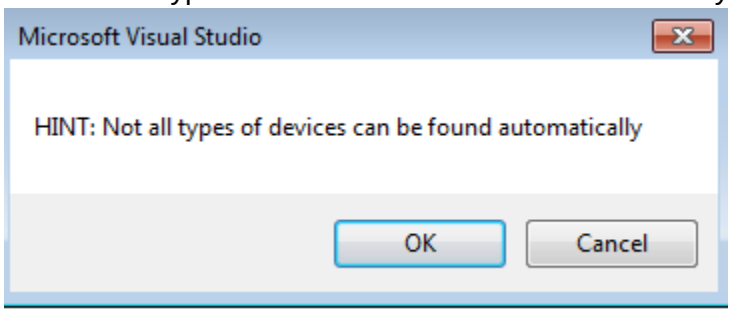
4. If the platform differs from current platform, change the solution platform.



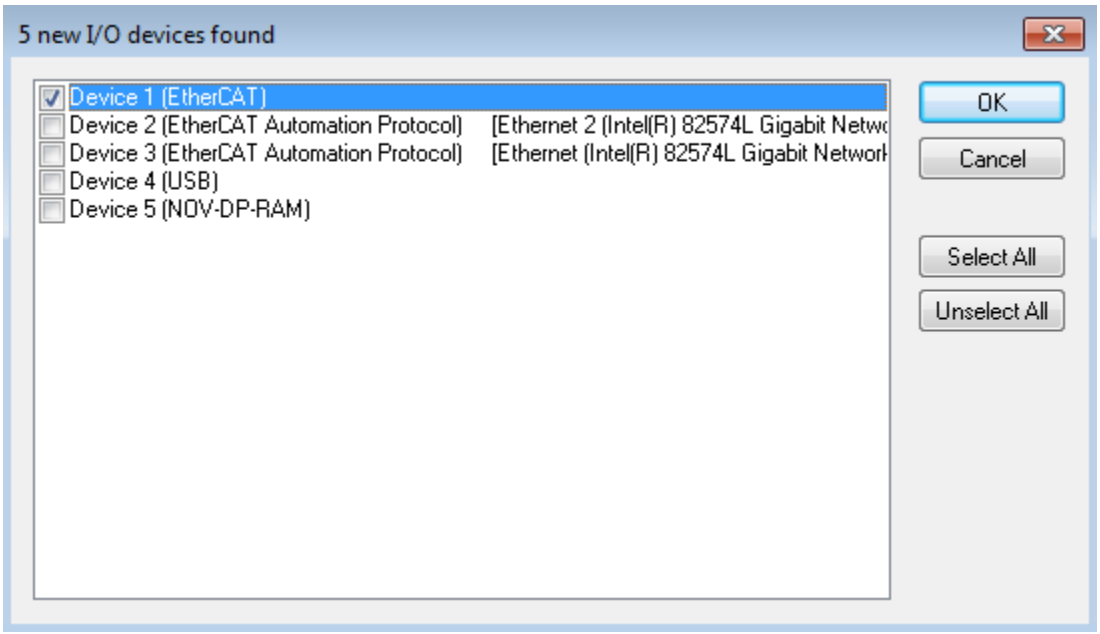
5. Right click on Devices in the Solution Explorer, and select “Scan…”



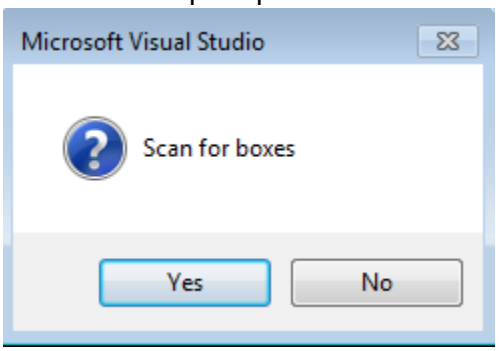
6. Accept box “Not all types of devices can be found automatically”



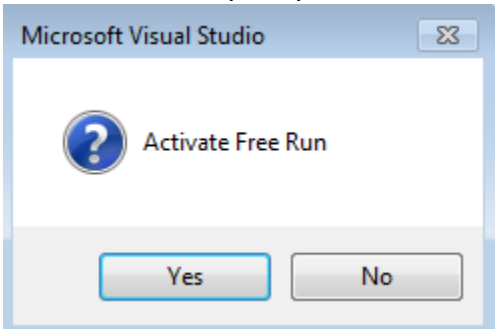
7. Select PLC Device



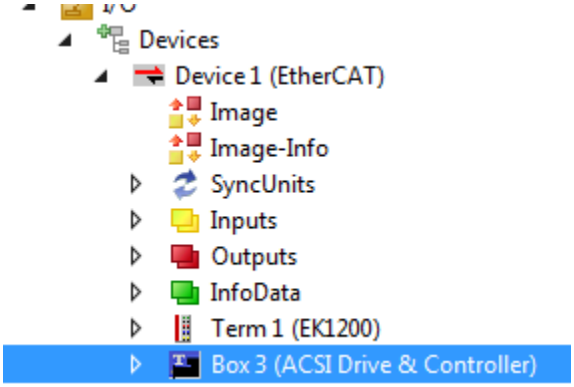
8. Scan for boxes when prompted



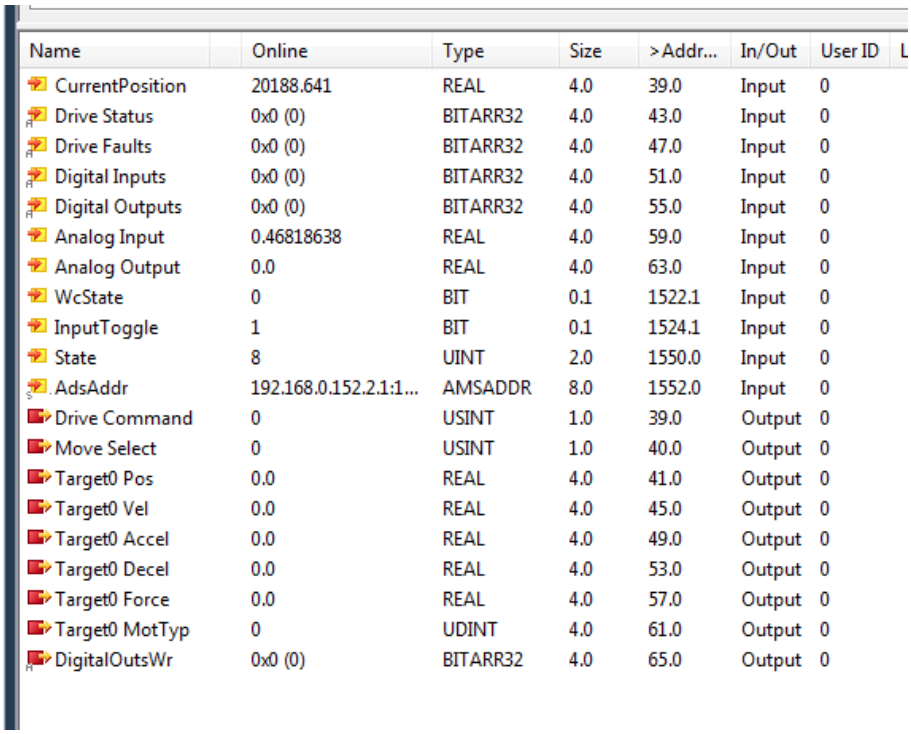
9. Activate Free Run when prompted



10. Verify ACSI Box was added to Device 1

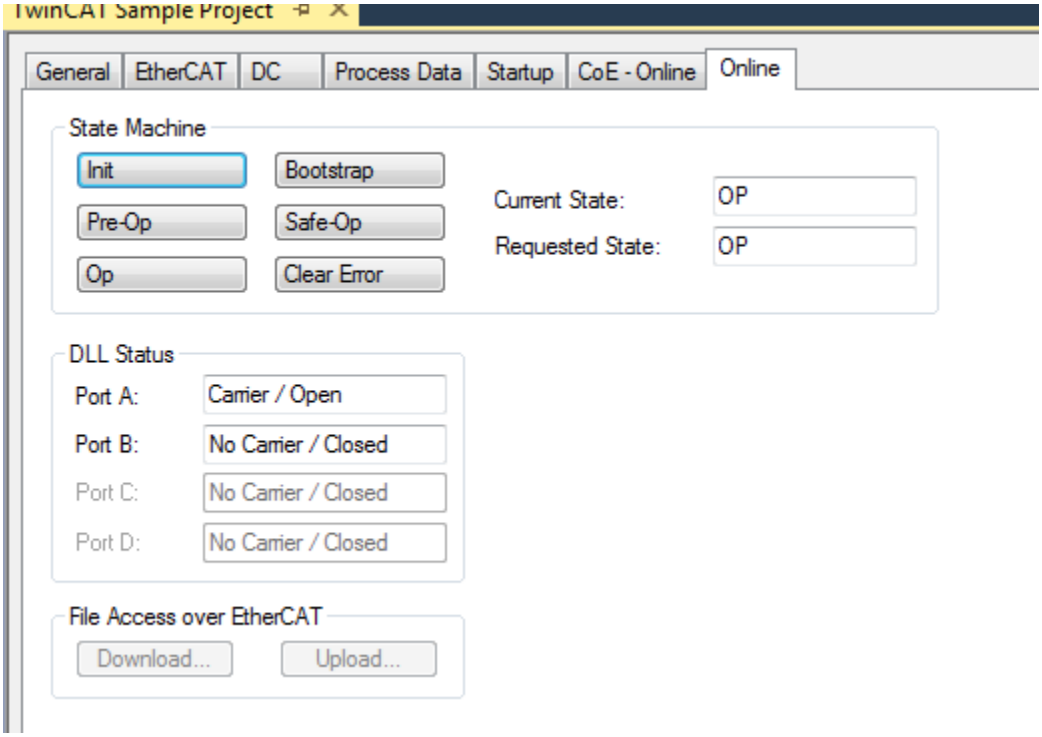


11. Double click on ACSI Box and verify “InputToggle” bit continuously toggles between ‘0’ and ‘1’



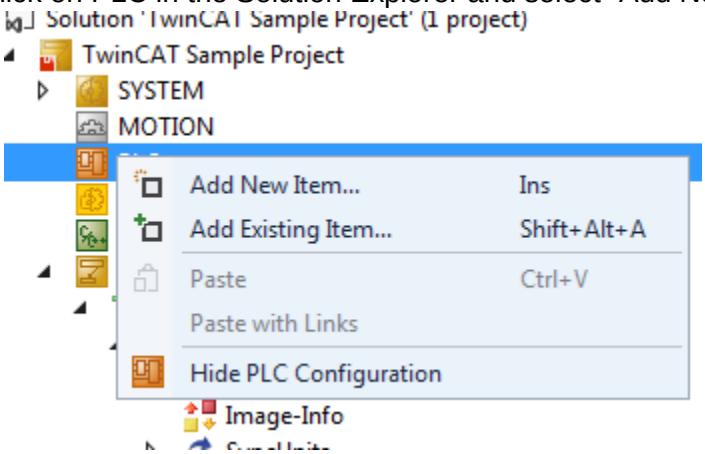
Name	Online	Type	Size	>Addr...	In/Out	User ID
CurrentPosition	20188.641	REAL	4.0	39.0	Input	0
Drive Status	0x0 (0)	BITARR32	4.0	43.0	Input	0
Drive Faults	0x0 (0)	BITARR32	4.0	47.0	Input	0
Digital Inputs	0x0 (0)	BITARR32	4.0	51.0	Input	0
Digital Outputs	0x0 (0)	BITARR32	4.0	55.0	Input	0
Analog Input	0.46818638	REAL	4.0	59.0	Input	0
Analog Output	0.0	REAL	4.0	63.0	Input	0
WcState	0	BIT	0.1	1522.1	Input	0
InputToggle	1	BIT	0.1	1524.1	Input	0
State	8	UINT	2.0	1550.0	Input	0
AdsAddr	192.168.0.152.2.1:1...	AMSADDR	8.0	1552.0	Input	0
Drive Command	0	USINT	1.0	39.0	Output	0
Move Select	0	USINT	1.0	40.0	Output	0
Target0 Pos	0.0	REAL	4.0	41.0	Output	0
Target0 Vel	0.0	REAL	4.0	45.0	Output	0
Target0 Accel	0.0	REAL	4.0	49.0	Output	0
Target0 Decel	0.0	REAL	4.0	53.0	Output	0
Target0 Force	0.0	REAL	4.0	57.0	Output	0
Target0 MotTyp	0	UDINT	4.0	61.0	Output	0
DigitalOutsWr	0x0 (0)	BITARR32	4.0	65.0	Output	0

Troubleshooting tip: If values do not update, or ACSI Box Online state does not go to “OP”, please reference the EtherCAT User’s Guide section about updating EEPROM through TwinCAT.

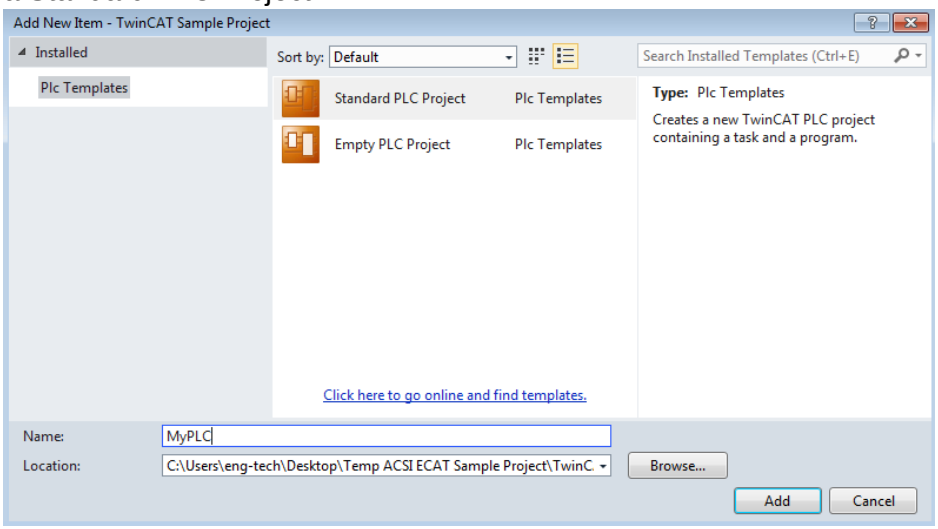


Installing ACSI TwinCAT Library

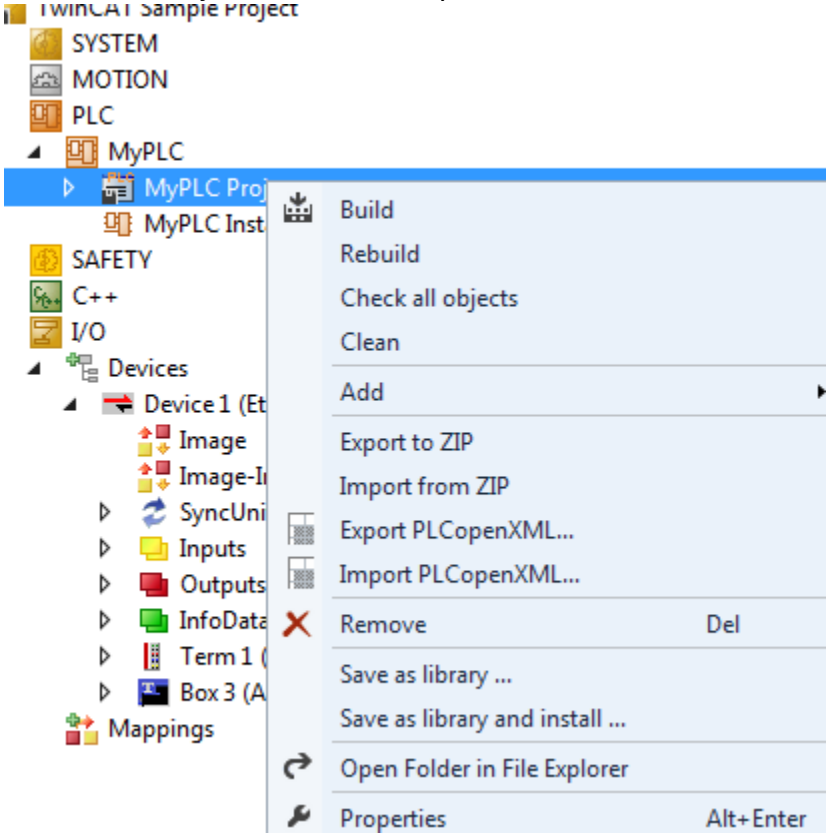
1. Right click on PLC in the Solution Explorer and select “Add New Item”



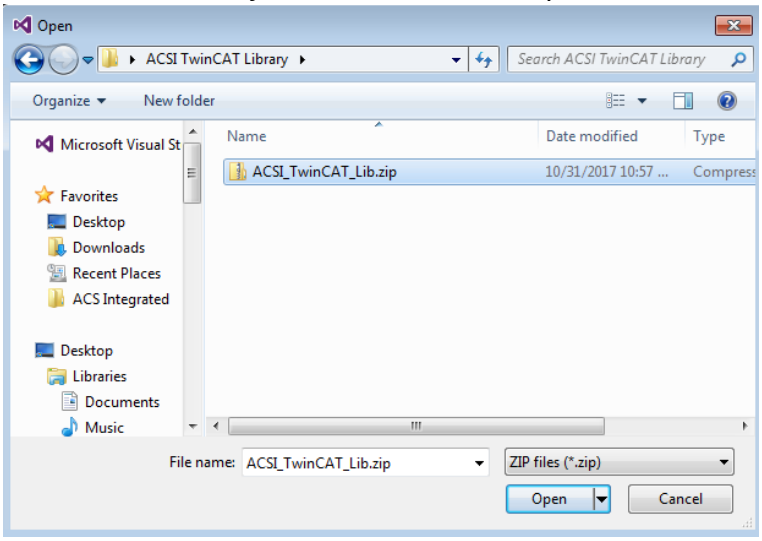
2. Create a Standard PLC Project



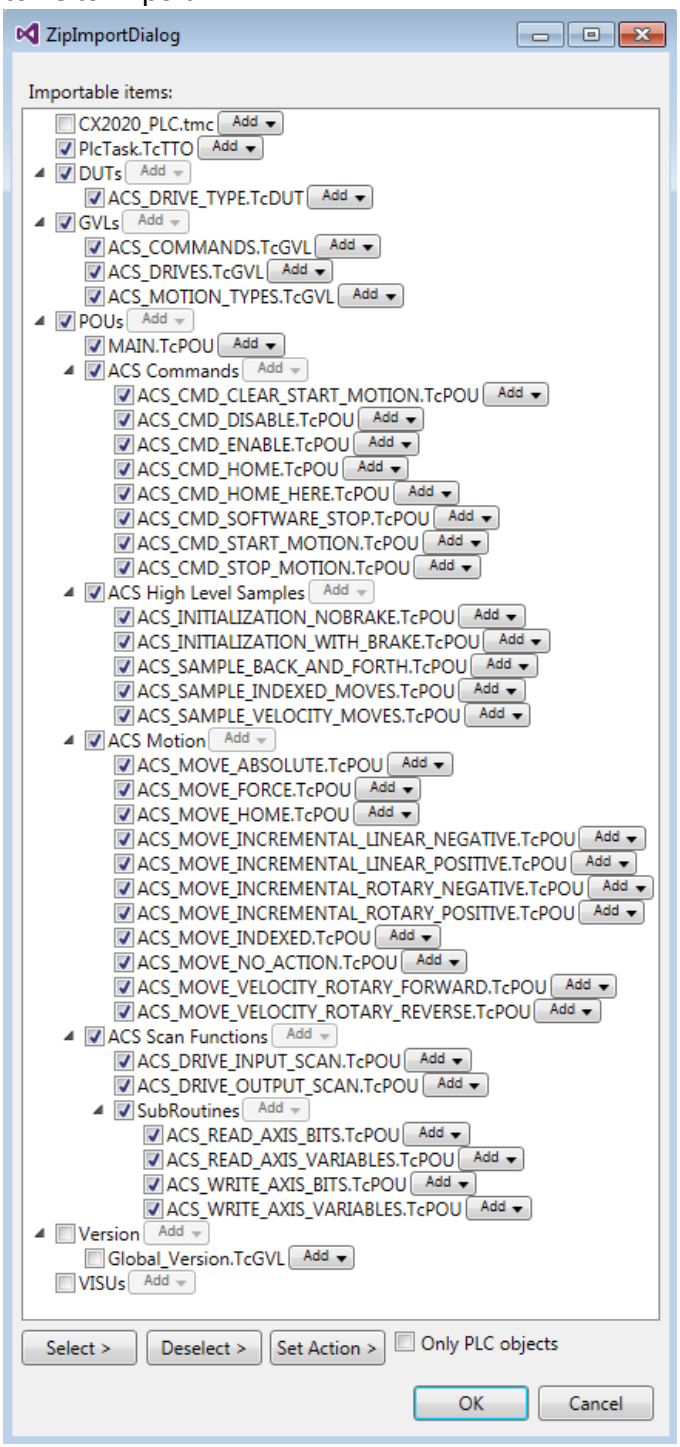
3. Right click on PLC Project and select “Import from ZIP”



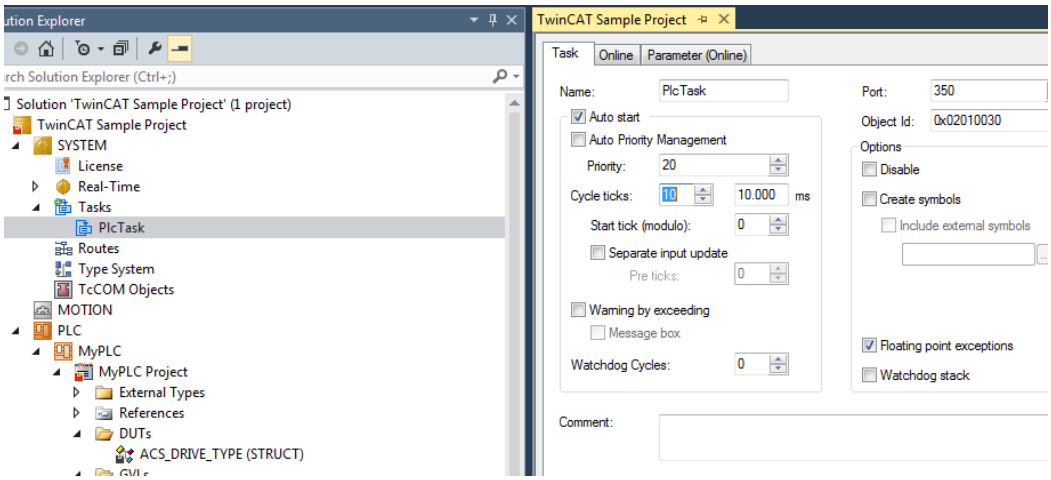
4. Browse to TwinCAT Library ZIP file and select “Open”



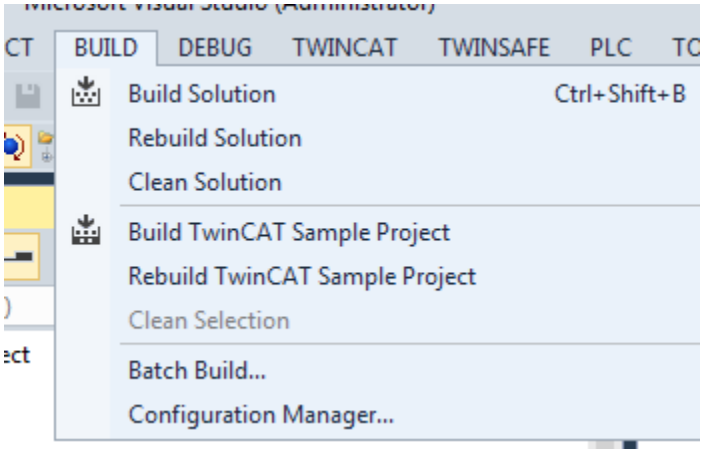
5. Select Items to Import



6. Verify PlcTask Cycle ticks set at 10.000 ms (Drive should not be written to faster than 10ms)



7. Select BUILD->Rebuild Solution



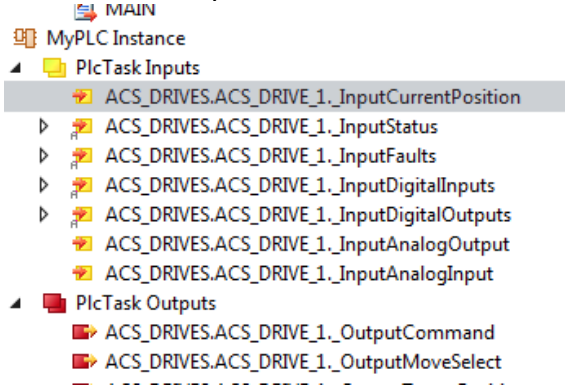
Resolve any errors with build (these typically include duplicate PlcTasks, Main functions, etc.)

Assign PlcTask Inputs and Outputs to Drive

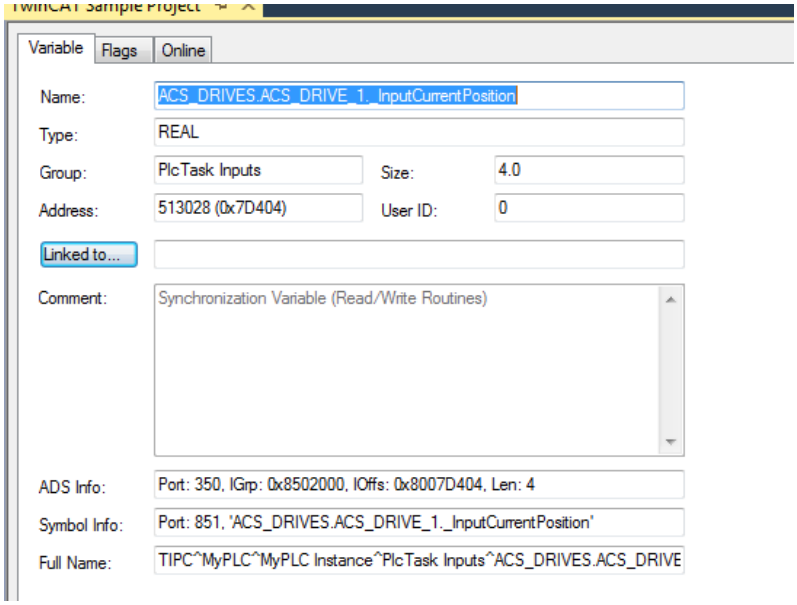
ACS Drives are defined in the ACS_DRIVES GVL. Initially, just a single (ACS_DRIVE_1) is defined, but the idea is that multiple axis definitions would be defined in this GVL for systems that contain multiple ACS Drives.

Under PlcTask Inputs in the Solution Explorer

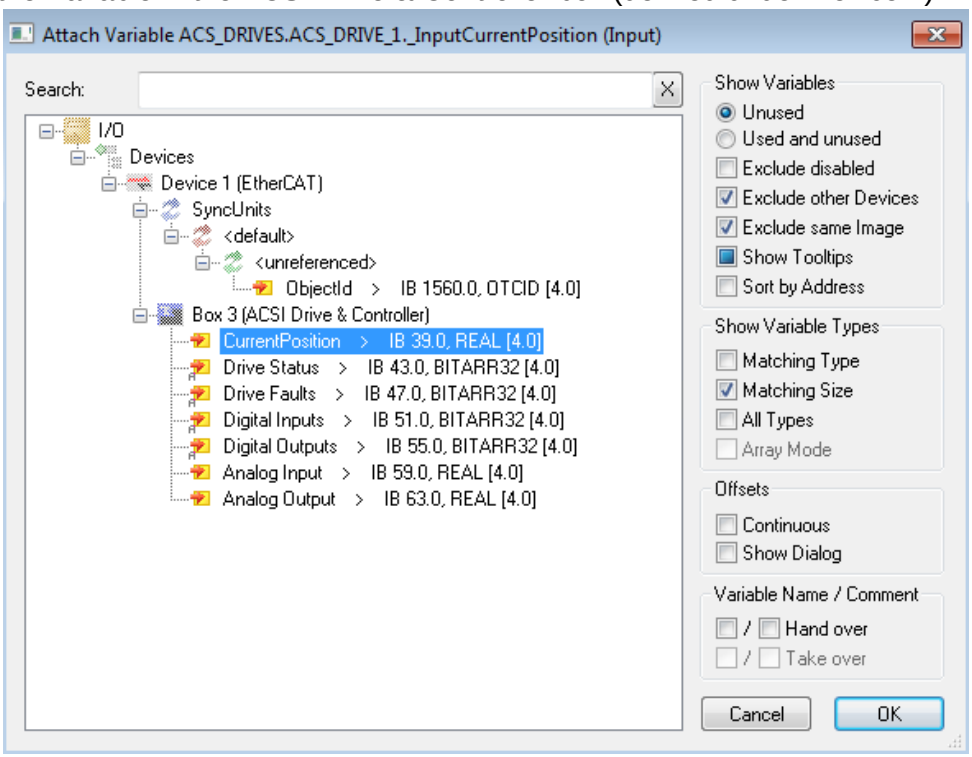
1. Double click on each Input



2. Select the “Linked to…” button



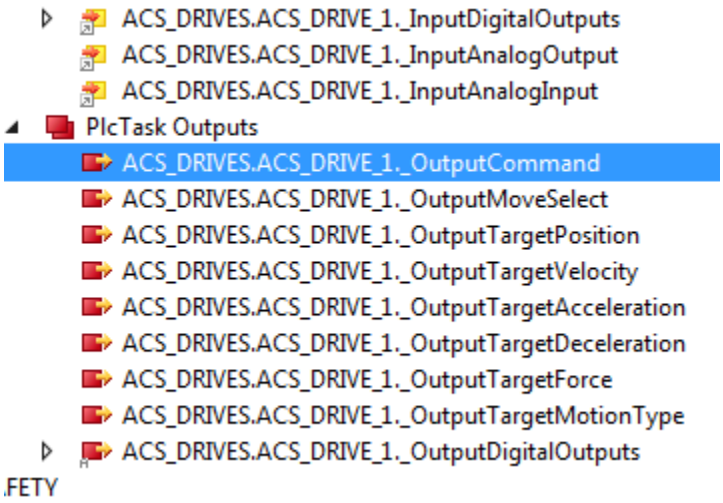
3. Select the Variable in the ACSI Drive & Controller box (defined under Device 1)



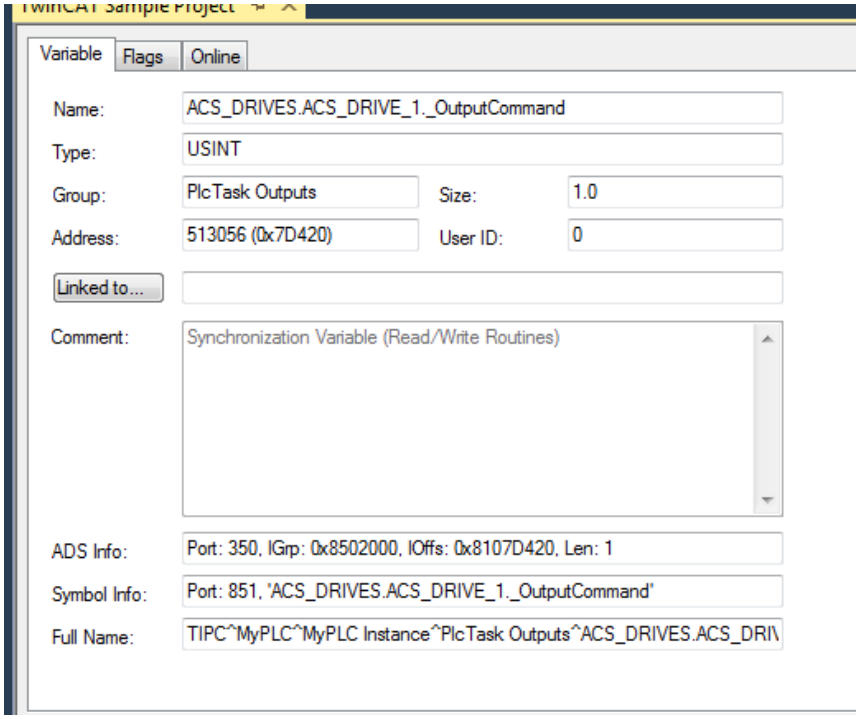
4. Use the following mapping
    a. ACS_DRIVES.ACS_DRIVE_1.InputCurrentPosition->CurrentPosition
    b. ACS_DRIVES.ACS_DRIVE_1.InputStatus->Drive Status
    c. ACS_DRIVES.ACS_DRIVE_1.InputFaults->Drive Faults
    d. ACS_DRIVES.ACS_DRIVE_1.InputDigitalInputs->Digital Inputs
    e. ACS_DRIVES.ACS_DRIVE_1.InputDigitalOutputs->Digital Outputs
    f. ACS_DRIVES.ACS_DRIVE_1.InputAnalogOutput->Analog Output
    g. ACS_DRIVES.ACS_DRIVE_1.InputAnalogInput->Analog Input

Under PlcTask Outputs in the Solution Explorer

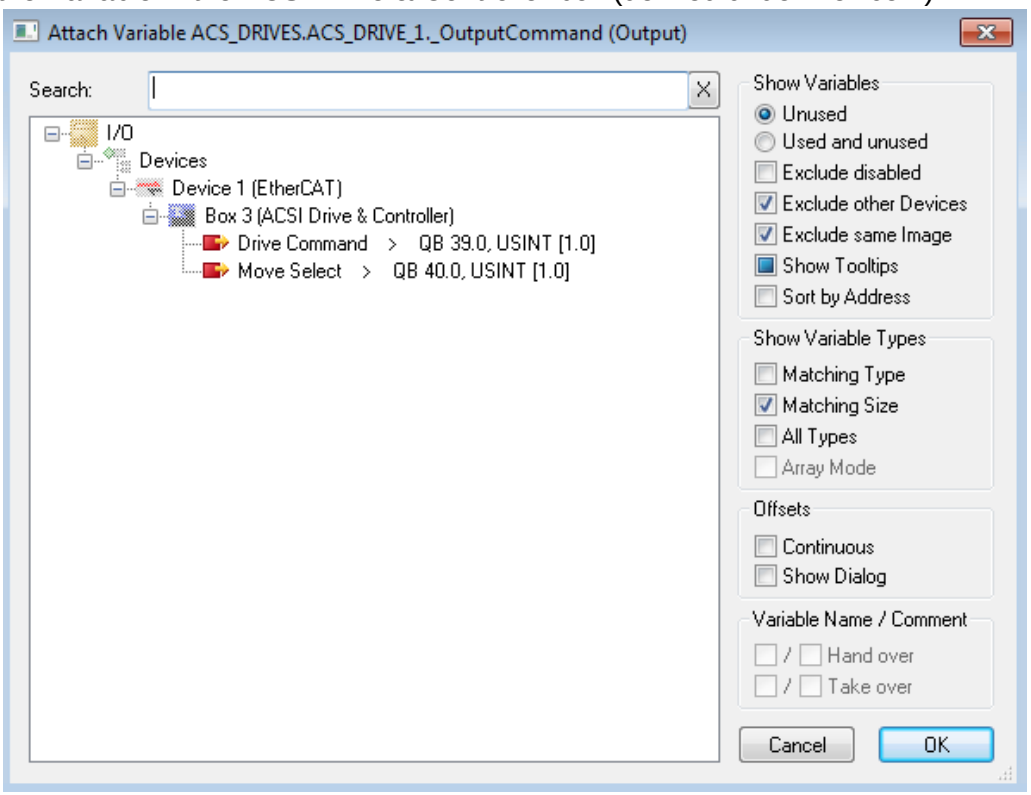
1. Double click on each Output



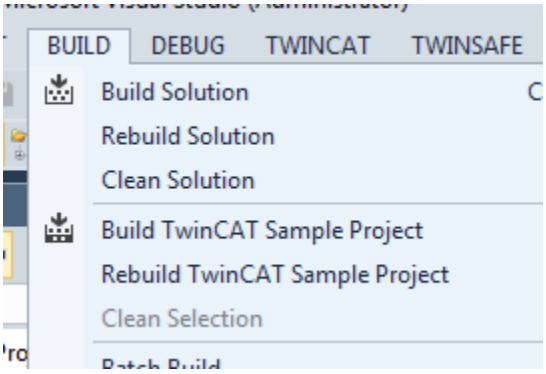
2. Select the “Linked to…” button



3. Select the Variable in the ACSI Drive & Controller box (defined under Device 1)

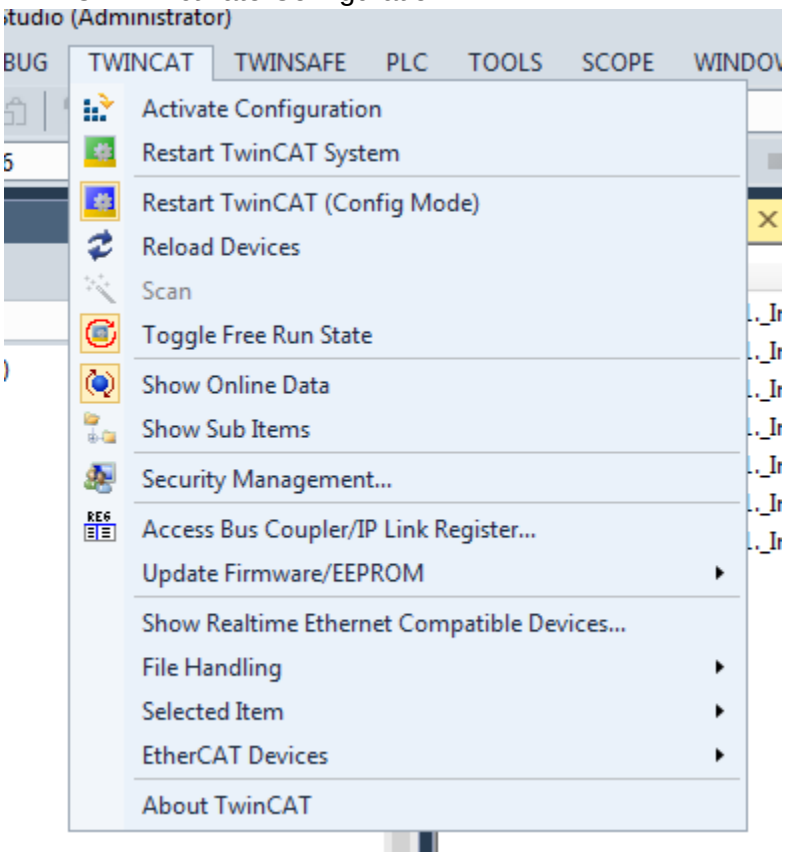


4. Use the following mapping
    a. ACS_DRIVES.ACS_DRIVE_1.OutputCommand->Drive Command
    b. ACS_DRIVES.ACS_DRIVE_1.OutputMoveSelect->Move Select
    c. ACS_DRIVES.ACS_DRIVE_1.OutputTargetPosition->Target0 Pos
    d. ACS_DRIVES.ACS_DRIVE_1.OutputTargetVelocity->Target0 Vel
    e. ACS_DRIVES.ACS_DRIVE_1.OutputTargetAcceleration->Target0 Accel
    f. ACS_DRIVES.ACS_DRIVE_1.OutputTargetDeceleration->Target0 Decel
    g. ACS_DRIVES.ACS_DRIVE_1.OutputTargetForce->Target0 Force
    h. ACS_DRIVES.ACS_DRIVE_1.OutputTargetMotionType->Target0 MotTyp
    i. ACS_DRIVES.ACS_DRIVE_1.OutputDigitalOutputs->DigitalOutsWr
5. Rebuild Solution

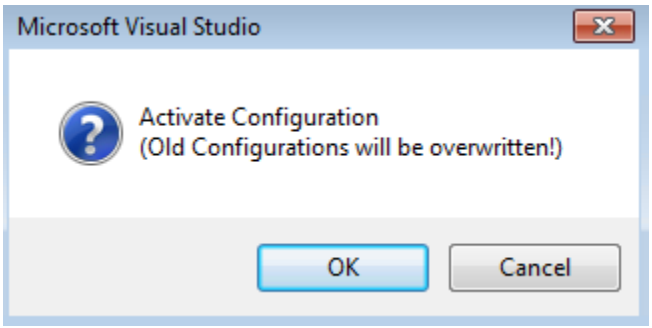


6. Test PLC Inputs

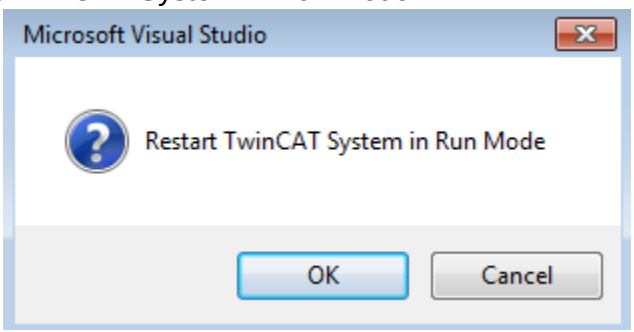
7. Select TWINCAT->Activate Configuration



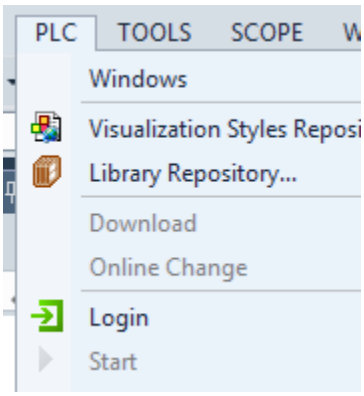
8. Accept Activation



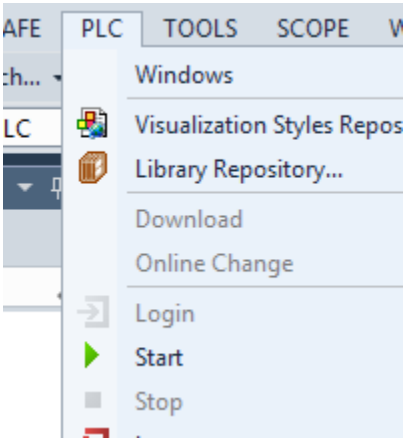
9. Restart TwinCAT System in Run Mode



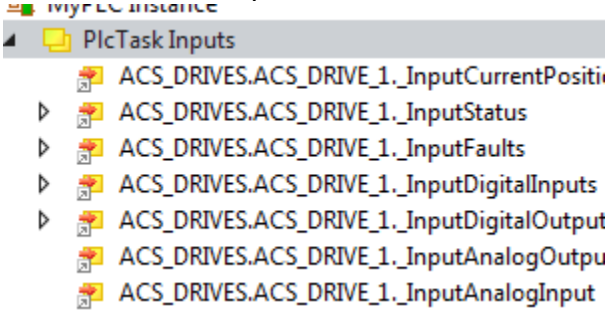
10. Log Into PLC



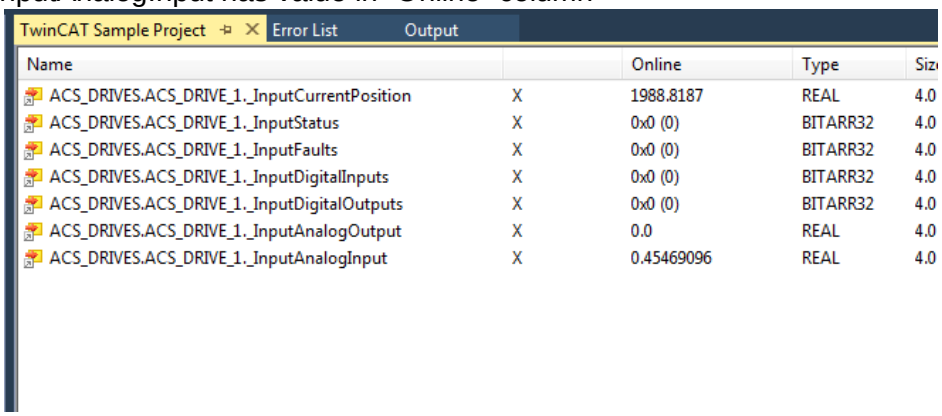
11. Start PLC



12. Double click on PlcTask Inputs



13. Verify InputAnalogInput has value in “Online” column

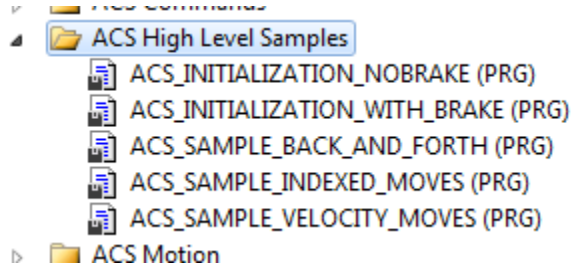


ACS TwinCAT Library Structure

The following sections outline the various parts of the library and how they interact with each other.

ACS High Level Samples

The Library includes a number of high level samples that encapsulate an entire single axis program. They include the Input Scan, some processing, and an Output Scan.



Also included are examples of initialization routines. These are entitled “ACS_INTIALIZATION_NOBRAKE”, and “ACS_INITIALIZATION_WITH_BRAKE”. These two different programs show that, depending on the hardware configuration of the drive and purpose, the initialization routines for drives could differ.

The primary difference between the NOBRAKE and WITH_BRAKE initializations is the check for the Brake Not Active status bit. We do not want to command motion with the brake enabled. This could cause false homing, I2T faults, or position errors.

```
//Only if there are no faults...
IF Axis.InputFaultActive < 1 THEN

    //If we are not enabled, then enable
    IF Axis.InputStatusDriveEnabled < 1 THEN
        ACS_CMD_ENABLE(Axis:=Axis);

    //If we are enabled, but not homed, then home
    ELSE
        IF Axis.InputStatusDriveHomed < 1 THEN
            ACS_CMD_HOME(Axis:=Axis);
        END_IF
    END_IF

END_IF
```

```
//Only if there are no faults...
IF Axis.InputFaultActive < 1 THEN

    //If we are not enabled, then enable
    IF Axis.InputStatusDriveEnabled < 1 THEN
        ACS_CMD_ENABLE(Axis:=Axis);

    ELSE
        //If we are enabled, but not homed, then home
        IF Axis.InputStatusDriveHomed < 1 THEN
            //Brake is turned off after a period of time while t
            IF Axis.InputStatusDriveBrakeNotActive = 1 THEN
                ACS_CMD_HOME(Axis:=Axis);
            END_IF
        END_IF
    END_IF

END_IF
```

In some installations, such as rotary installations, you may not require homing as part of the initialization. If the drive is configured in rotary mode (accomplished in Tolomatic Motion Interface), homing is optional. See the ACS_SAMPLE_VELOCITY_MOVES program. The initialization routine is simply enabling the drive if there are no faults and it is disabled.

Command Register and Fault Clearing

ACS drives are command edge triggered. This means that the drive will only use Outputs when it detects a change in the Output Command register. Motion profiles can be changed continuously as long as the command register remains the same. As soon as the command register is updated, the drive will execute the command.

The command register is polled at a periodic rate internally to the drive. If commands are changed too quickly, the drive may not see a command, and not execute the command. This is especially important to note when making very fast moves, and clearing faults.

Some faults are cleared by disabling and re-enabling the drive. (See TMI Users Guide for further detail about faults). In practical terms, this means that the program should disable the drive for at least 20ms (longer is better) before re-enabling the drive.

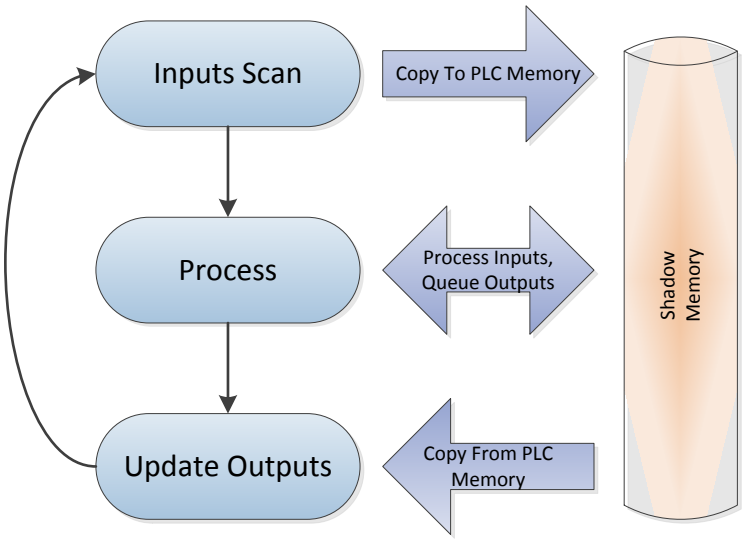
ACS Interface Functions

The PLC program will interface with the ACS drive using three categories of functions

1. Scan Functions
2. Motion Functions
3. Command Functions

ACS Scan Functions

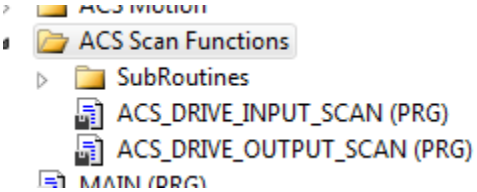
A typical PLC program will contain an Input Scan at the beginning, a Processing section, and an Output Scan (or update) section.



For the Input Scan routine, the library provides a function called “ACS_DRIVE_INPUT_SCAN” which updates all of the Inputs from the drive into memory that can be manipulated by the processing functions.

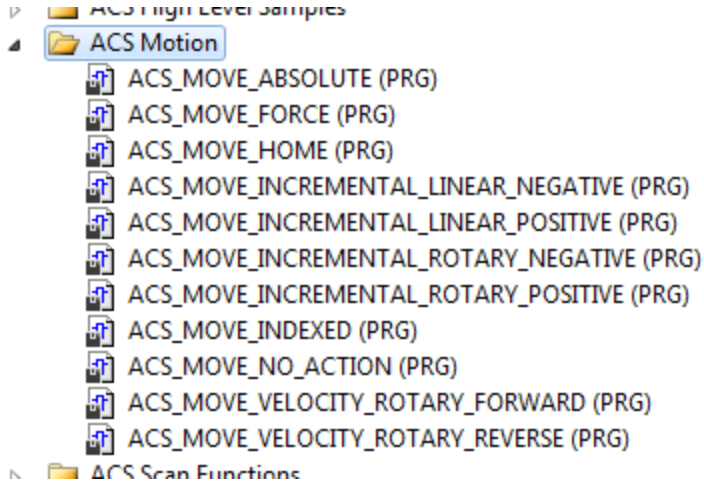
For the Output Scan routine, the library provides a function called “ACS_DRIVE_OUTPUT_SCAN” which copies all of the memory that was manipulated by the processing functions back to the drive.

There are also a number of sub routines that the INPUT_SCAN and OUTPUT_SCAN use, however the PLC program should not need to access these directly.



Motion Functions

The primary purpose for the motion functions is to prime the drive outputs for a specific motion function. These are typically different types of move. Note that some moves are specific for Rotary applications and some are for linear applications. Rotary applications do not require homing, and need to be configured in Tolomatic Motion Interface.



The motion functions require a reference to the target axis, as well as additional motion profile parameters specific for the move type. An Absolute Move example is below.

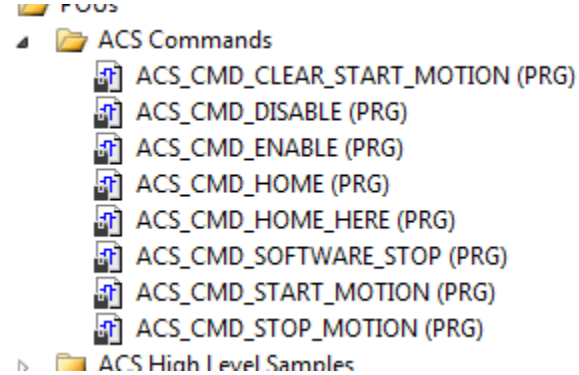
```
ACS_MOVE_ABSOLUTE(Axis:=Axis,
                  ProfilePosition:=ExtendPosition,
                  ProfileVelocity:=ProfileVelocity,
                  ProfileAcceleration:=ProfileAccelDecel,
                  ProfileDeceleration:=ProfileAccelDecel,
                  ProfileForce:=ProfileForce);
```

Some motion functions, such as the Home function, and Indexed function, use predefined motion profiles (in Tolomatic Motion Interface). These do not require the PLC to define the motion profiles.

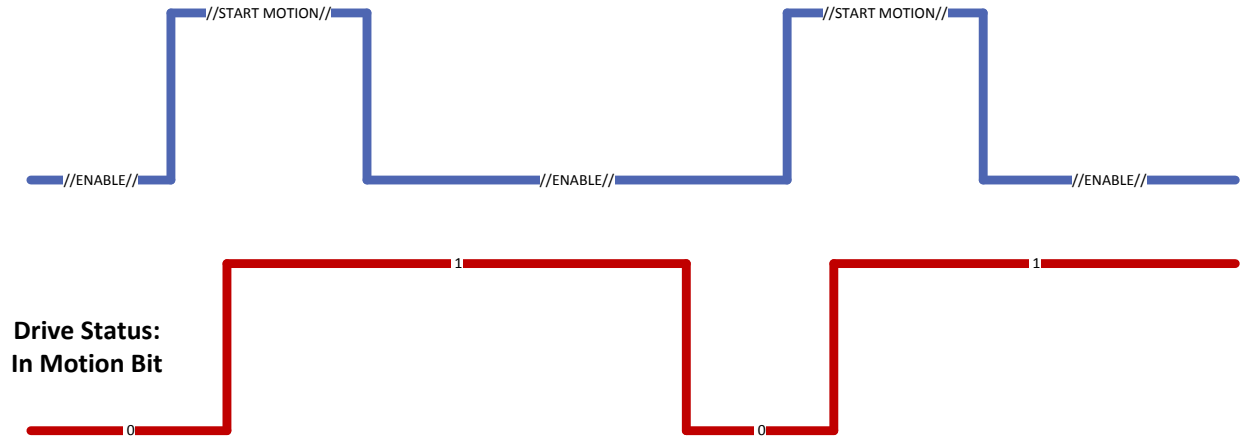
Execution of these motion functions do not cause actual motion to occur. In order to start motion, the user must perform an ACS Command.

ACS Command Functions

The ACS Commands defined in the library allow the user to send specific commands to the drive. These include starting motion, homing, performing the ESTOP (Software Stop) action (defined in Tolomatic Motion Interface), etc.



It is important to note that the ACS Drive is Edge Triggered, meaning that the drive must detect a change in the Command Register in order to execute a command. See the following example.

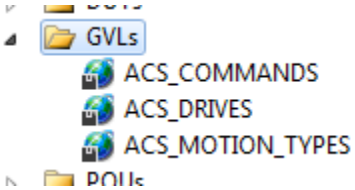


This also can lead to race conditions on the Command register. The user must use caution to prevent multiple different concurrent programs from sending commands to the drive during the same scan cycle. Only one command will be executed. The drive also polls the command register at an approximate 10ms rate. Commands updating faster than this rate will be ignored.

It is better to design the PLC program to prevent writes to the Drive faster than 10ms.

ACS Drives Global Variable Lists

The library contains three Global Variable Lists (GVLs). Two of these lists (ACS_COMMANDS and ACS_MOTION_TYPES) are constant variables and used in library functions. The third, ACS_DRIVES, is meant to be a container for ACS Drive Axes.

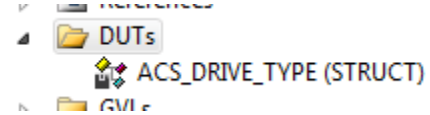


Multiple drive axes can be defined in this list, and the user can access them directly from programs. Most functions require a reference to an ACS_DRIVE_TYPE variable.

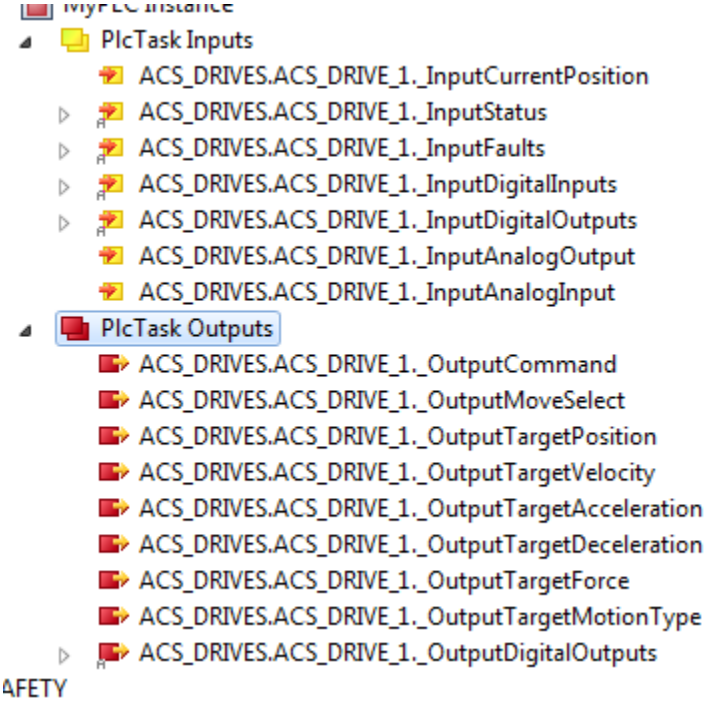
```
{attribute 'qualified_only'}
VAR_GLOBAL
    //Global GVL for drives
    ACS_DRIVE_1     : ACS_DRIVE_TYPE;
END_VAR
```

ACS Data Types

There is one data type defined in the ACS TwinCAT Library. The ACS_DRIVE_TYPE is a structure that contains mapping to input and output registers for a specific drive, as well as a shadow memory area for library function processing.



When a variable is defined as ACS_DRIVE_TYPE, it creates PlcTask Inputs and PlcTask Outputs for the defined variable.



The user then is required to link these inputs and outputs to a defined I/O box

Additional information for the ACSI EtherCAT implementation can be found in:
3604-4185 ACSI Hardware User's Guide,
3600-4201 EtherCAT User's Guide,
3604-4184 Tolomatic Motion Interface User's Guide.

See tolomatic.com for the most up-to-date technical information.

Tolomatic
EXCELLENCE *IN MOTION*®

COMPANY WITH QUALITY SYSTEM CERTIFIED BY DNV GL = ISO 9001 =
Certified site: Hamel, MN

USA
3800 County Road 116
Hamel, MN 55340, USA
Phone: (763) 478-8000
Fax: (763) 478-8080
Toll-Free: **1-800-328-2174**
sales@tolomatic.com
www.tolomatic.com

CHINA
Tolomatic Automation Products (Suzhou) Co. Ltd.
(ServoWeld® inquiries only)
No. 60 Chuangye Street, Building 2
Huqiu District, SND Suzhou
Jiangsu 215011 - P.R. China
Phone: +86 (512) 6750-8506
Fax: +86 (512) 6750-8507
ServoWeldChina@tolomatic.com

EUROPE
Tolomatic Europe GmbH
Zeilweg 42
60439 Frankfurt am Main
Germany
Phone: +49 69-2045-7837
EuropeSales@tolomatic.com

Visit www.tolomatic.com for the most up-to-date technical information