**To run files**

1. Create twinCAT project and PLC
2. Run PLC
3. Run matlab files

\*\*IMPORTANT TO NOTE

* After running the PLC but before running MATLAB files make sure to activate robots, the below should be copied into the command window, activating and homing each robot

|  |
| --- |
| asm = NET.addAssembly(which('TwinCAT.Ads.dll'));  import TwinCAT.Ads.\*;    %create an ads client  adsClt=TwinCAT.Ads.TcAdsClient;    adsClt.Connect('192.168.0.111.1.1',851);    %get info of variables  info\_activate\_1=adsClt.ReadSymbolInfo('GVL.Robot1.Outputs.Robot\_Control.Activate');  info\_deactivate\_1=adsClt.ReadSymbolInfo('GVL.Robot1.Outputs.Robot\_Control.Deactivate');  info\_home\_1=adsClt.ReadSymbolInfo('GVL.Robot1.Outputs.Robot\_Control.Home');    info\_activate\_2=adsClt.ReadSymbolInfo('GVL.Robot2.Outputs.Robot\_Control.Activate');  info\_deactivate\_2=adsClt.ReadSymbolInfo('GVL.Robot2.Outputs.Robot\_Control.Deactivate');  info\_home\_2=adsClt.ReadSymbolInfo('GVL.Robot2.Outputs.Robot\_Control.Home');  adsClt.WriteAny(info\_deactivate\_1.IndexGroup,info\_deactivate\_1.IndexOffset,false);  adsClt.WriteAny(info\_activate\_1.IndexGroup,info\_activate\_1.IndexOffset,true);  adsClt.WriteAny(info\_home\_1.IndexGroup,info\_home\_1.IndexOffset,true);  adsClt.WriteAny(info\_deactivate\_2.IndexGroup,info\_deactivate\_2.IndexOffset,false);  adsClt.WriteAny(info\_activate\_2.IndexGroup,info\_activate\_2.IndexOffset,true);  adsClt.WriteAny(info\_home\_2.IndexGroup,info\_home\_2.IndexOffset,true); |

* **Running MATLAB files**
  + After the robots have been activated, you can run the main matlab files
  + When running matlab file it will first do some setup commands
  + You will want to either input 1 to run the main file which is an infiinite loop or 2 to deactivate the robots
    - If you input 1: it will first do some calibration and then run the main program on an infinite loop
    - If you input 2: it will deactivate the Robots
      * After it will ask if you want to Repeat: in which you should input any number except for 0 to end the program
* **\*\*FUTEK sensor specific info**
  + **%at the end of each run of the program copy this into the command window**
  + repo.DisconnectAllDevices();

**Force sensor control, z position and angle control**

Matlab files

|  |  |  |
| --- | --- | --- |
| File Name | Type | Notes |
| ForceFeedback\_ultrasound.m | main file | Reads and Sends data to twinCAT |
| PID\_angleFix.m | Function | Purpose: find error of angle |
| Variable: 7 values   * upper and lower limits of desired force\_x and force\_y for proper angle orientation   Values can be found through testing on a flat surface of alike material  \*\*refer to PID values.xlsx for some recommended values |
| PID\_zposFix.m | function | Purpose: find error of z-position |
| Variable: 5 values   * upper and lower limits of desired force\_x for proper z-position   Values can be found through testing and plotting force\_x  \*\*refer to PID values.xlsx for some recommended values |
| Offset\_fix.m | Calibration function | Purpose: find y and z offset due to angle change |
| Variable: 1 value   * Distance of probe   Values can be found by measuring height of the probe |
| Force\_correction.m | Calibration function | Purpose: fix raw force due to angle orientation |
| Variable: 2 values   * 2 sinusoidal equations   Values can be found by running Orientation\_forceoffset\_test.m to find the plot of the change in force offset from the true force and then fitting a sine wave of best fit |

TwinCAT files

|  |  |  |
| --- | --- | --- |
| File Name | Type | Notes |
| Project32.sln | Main project | Sets up the interface for communicating with EtherCAT devices |

**Haptic set up**

Matlab files

|  |  |  |
| --- | --- | --- |
| File Name | Type | Notes |
| ForceFeedback\_Haptic.m | Main file | Reads and Sends data to twinCAT  Reads and sends data to haptic device  \*\*\*\*For BOTA sensor |
| ForceFeedback\_Haptic\_futek.m | Main file | Reads and Sends data to twinCAT  Reads and sends data to haptic device  \*\*\*\*For Futek sensor |

TwinCAT files

|  |  |  |
| --- | --- | --- |
| File Name | Type | Notes |
| Project32.sln | Main project | Sets up the interface for communicating with EtherCAT devices |

**EXTRA**

TwinCAT Project 33 :: is the set up for two MECA’s

**How does Matlab communicate with TwinCAT?**

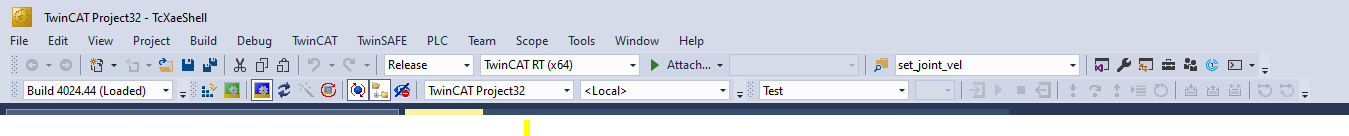
* The Twin CAT to Matlab communication is achieved through ADS communication

IMPORTANT FUNCTIONS

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Input | | Output |
| AdsClt.ReadSymbolInfo( ) | 1 | * Name of variable in PLC   \*\*make sure it matches name designated in PLC | Get Info of PLC variable   * Includes index group and index offset |
| AdsClt.WriteAny( ) | 3 | * Index group of variable * Index offset of variable * Value you would like variable to be changed to   \*\*for the new value:  Bool should be either true of false  REAL should be converted to single(new value) | Change a PLC variable value |
| AdsClt.ReadSymbol( ) | 1 | * Info of PLC variable | Get value of PLC variable |

**TwinCAT Setup**

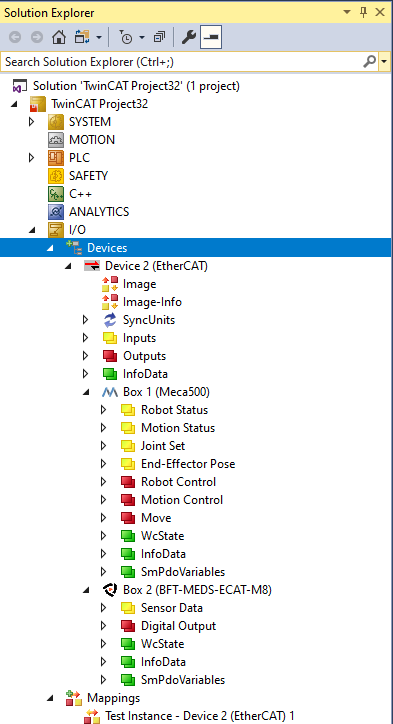
1. **Connect devices to TwinCAT**
   1. Open a new project
   2. Ensure the target system is on <Local>



* 1. In solution Explorer à I/O à Devices : Then right click à Scan

After the scan it should look similar to the image below

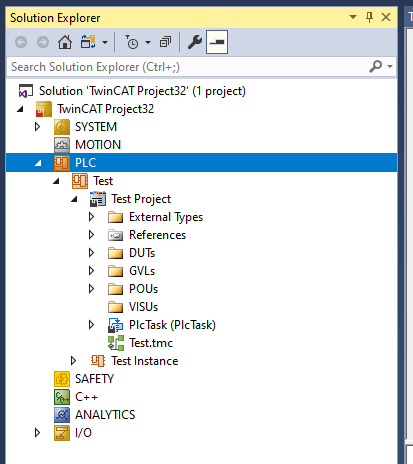
\*\* also you only need to add the EtherCAT device



* 1. In solution Explorer à PLC : right click à Add new item à Open Project

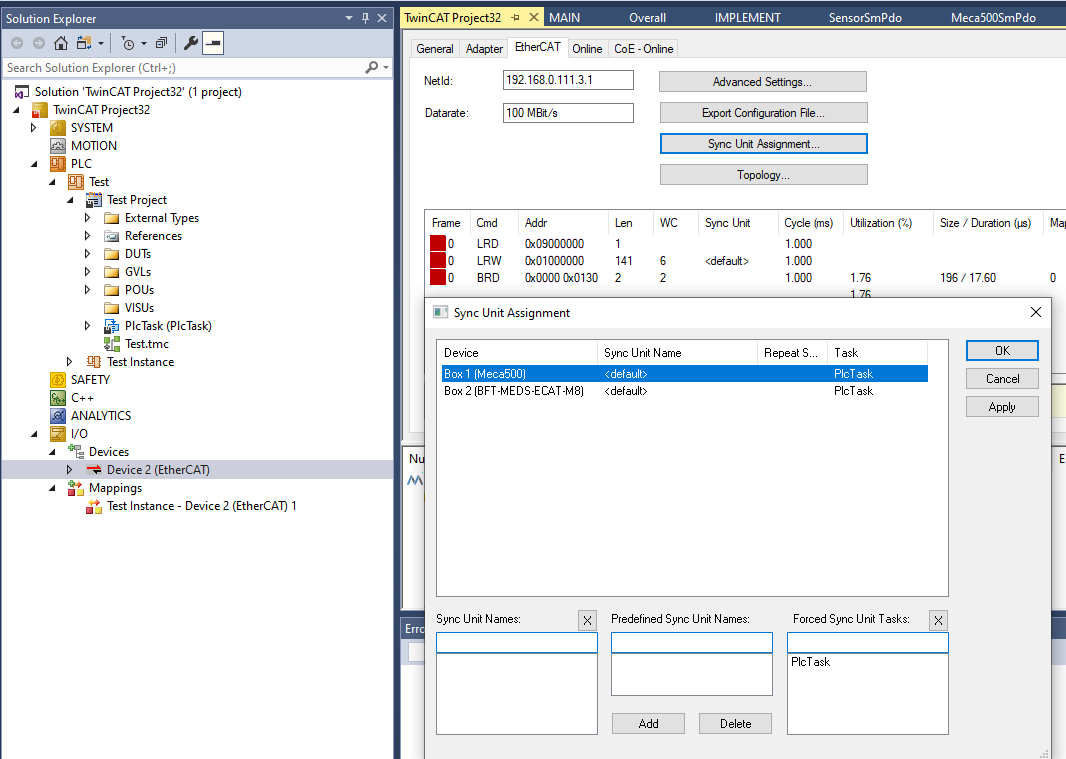
\*\*POUs is where the program will go

\*\*GVLs is where the global variables are stored



* 1. Select Device à EtherCAT tab à sync Unit Assignment à Add PLC task to the devices

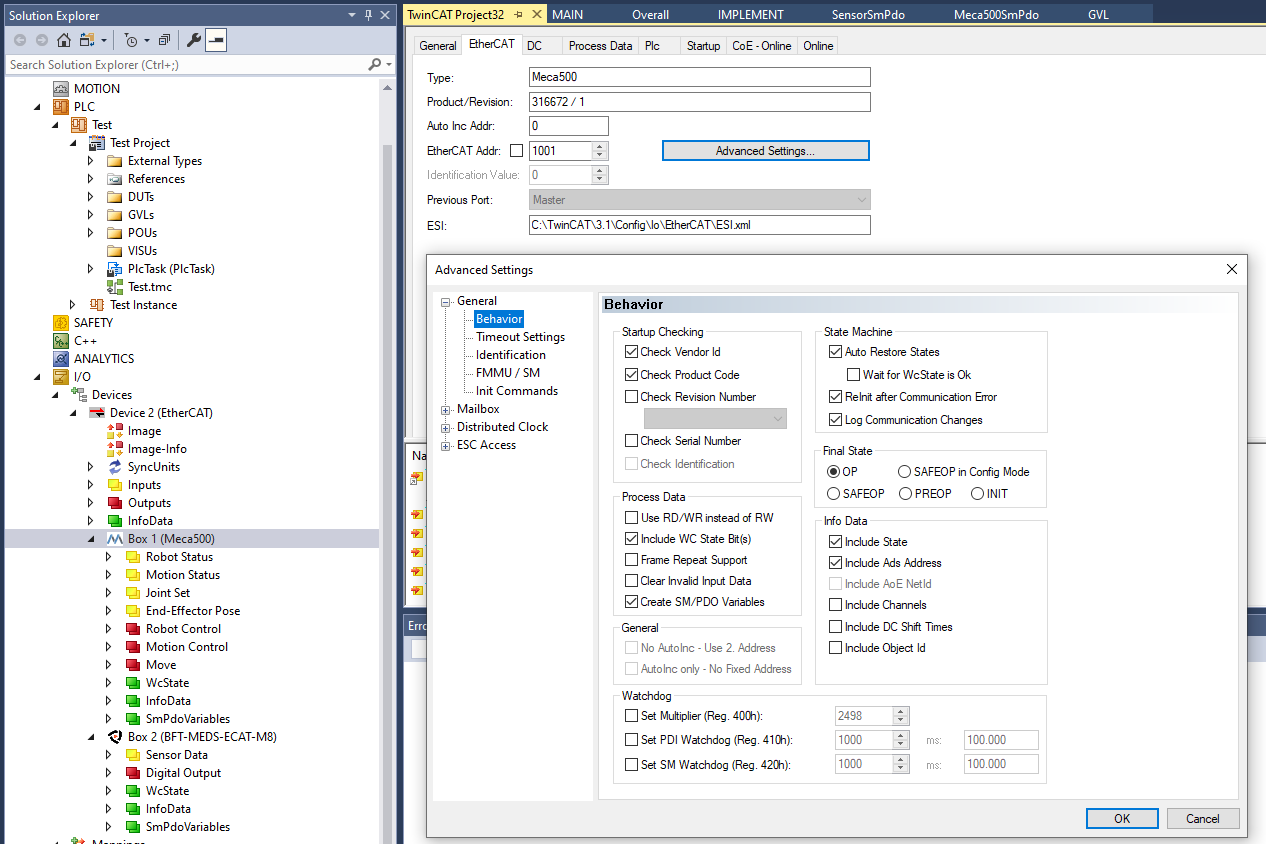
\*\* This connects the PLC program with the devices



* 1. Select the device connected (ie MECA500) --> EtherCat tab --> Advanced Settings --> select Create SM/PDO Variables

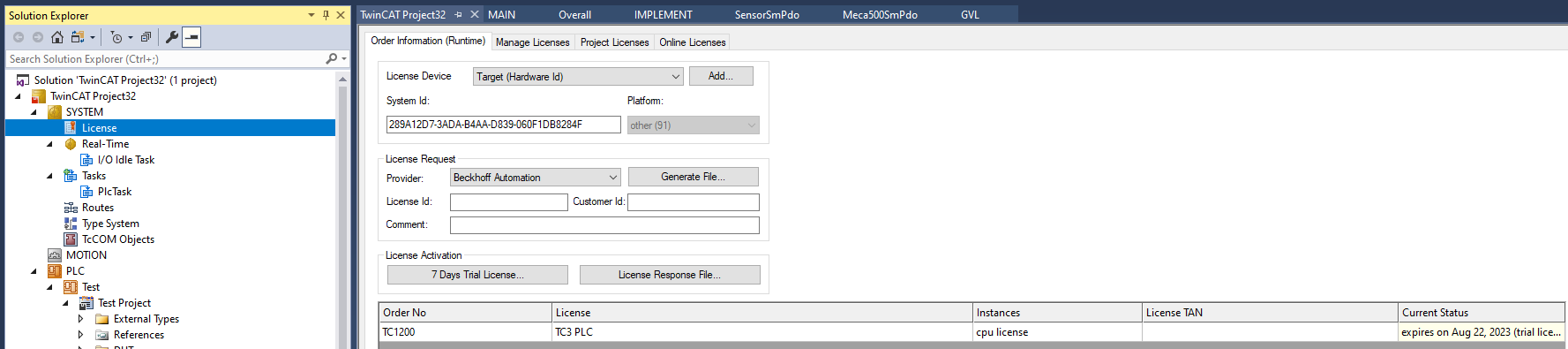
\*\* Do this for every device connected

\*\* This connects the variables from the connected devices



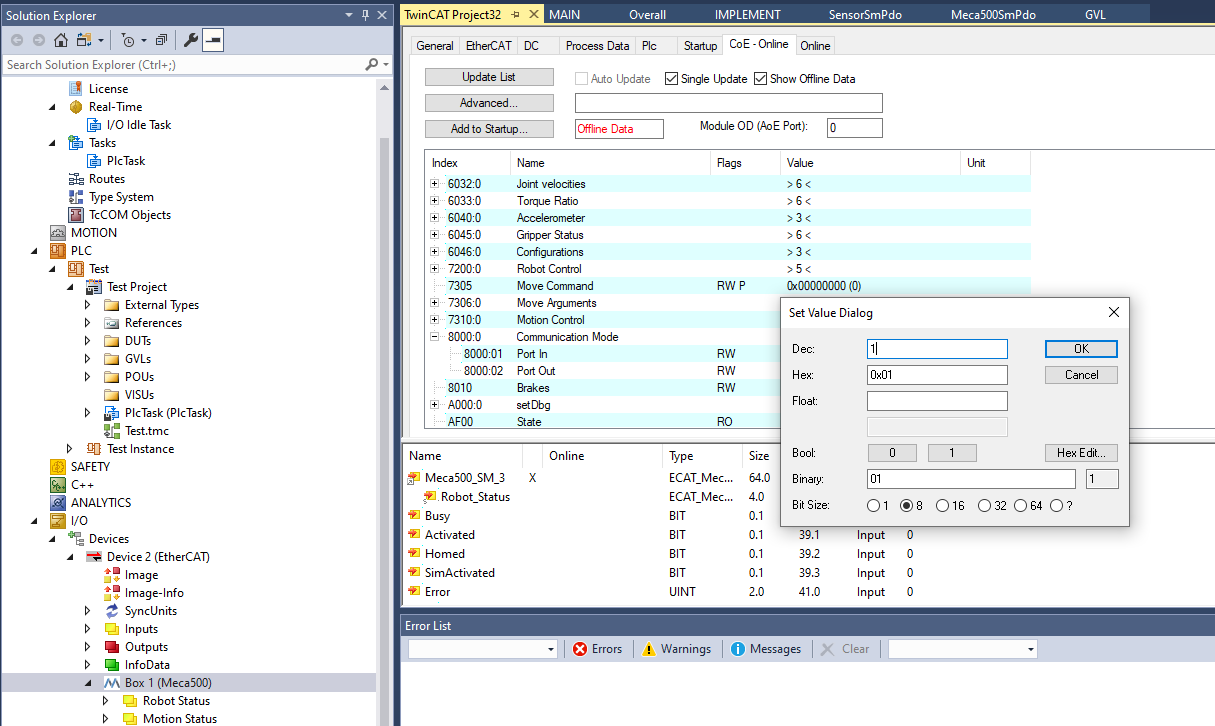
* 1. Update license : In system explorer à System à License

\*\*7 days trial license



**MECA specific information**

1. Switching between EtherCAT and Ethernet
   1. Once you have devices scanned onto an etherCAT project OR you can use a previous already made project ie Project 32
   2. Select MECA500 device --> CoE – Online tab --> Scroll down to 8000:0 --> maximize by clicking on the plus sign --> double click on 8000:01 --> change Dec from 2 to 1



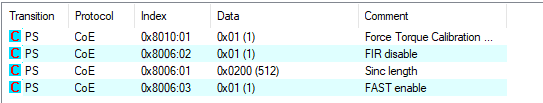
1. MECA commands for motion

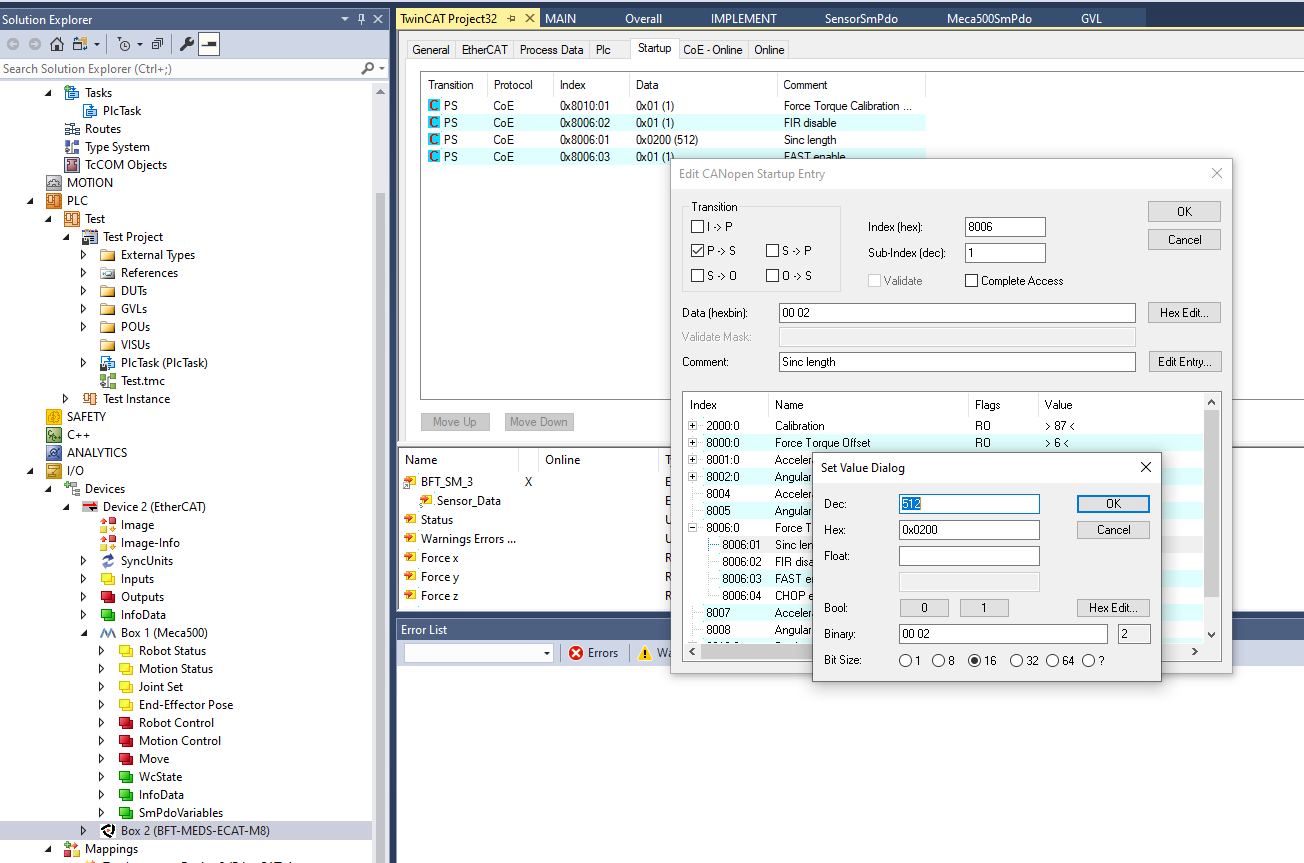
* To give a move command in matlab requires 11 values
  + [# # # # # # Move\_command Rand1 Rand2 1 1]
  + First 6 values are the values for the move
  + Move\_command: value that lets robot know which motion to do
  + Rand1 and Rand 2: random numbers for purposes of knowing when a new command is given and been completed
  + Last two numbers are letting the PLC know that a new command has been given

|  |  |
| --- | --- |
| MECA500 Move command | Type |
| 0 | No command |
| 1 | Move joints |
| 2 | Move pose |
| 3 | Move linear |
| 4 | Move linear relative to tool reference frame |
| 5 | Move linear relative to world reference frame |
| 6 | Add time delay for motion command |
| 7 | Set blending |
| 8 | Set joint velocity |
| 9 | Set joint acceleration |
| 10 | Set cartesian angular velocity |
| 11 | Set cartesian linear velocity |

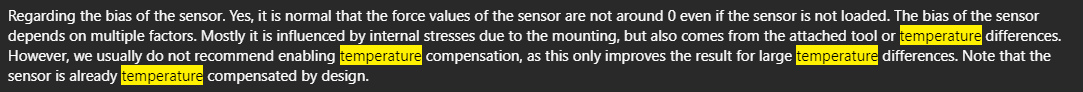
**BOTA specific information**

1. Setting proper startup command
   1. Once you have devices scanned onto an etherCAT project OR you can use a previous already made project ie Project 32
   2. Select BFT-MEDS-ECAT-M8 device --> Startup tab --> New --> find the parameter you want to add whenever sensor starts up--> double click on it --> change Dec value
   3. Repeat for the following settings shown in image



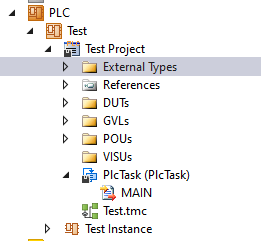


Temperature calibration is not turned on based on info sent by BOTA:



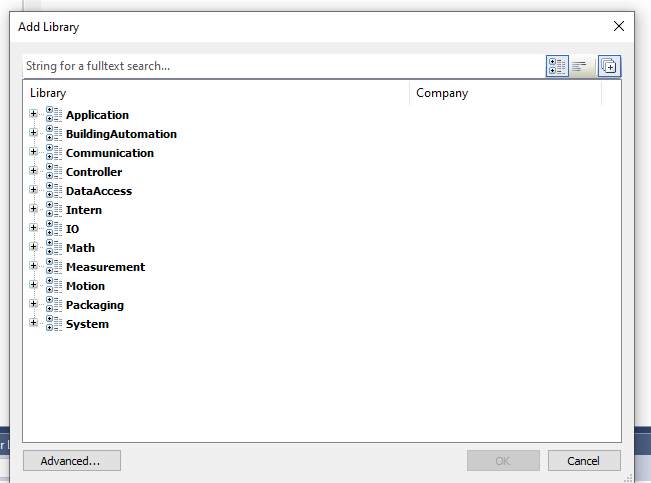
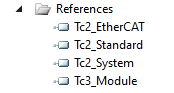
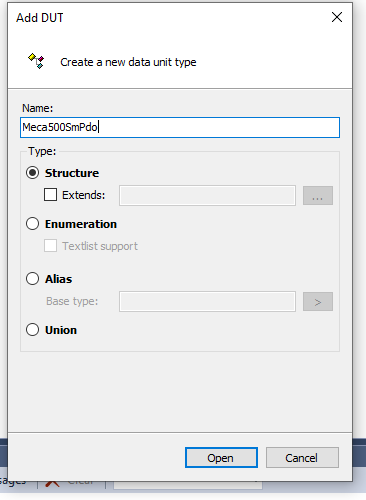
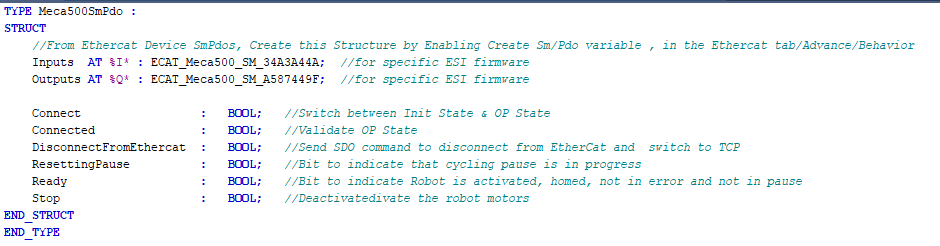
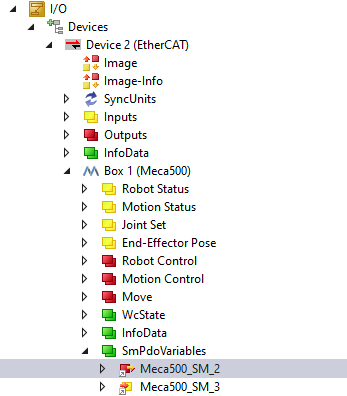
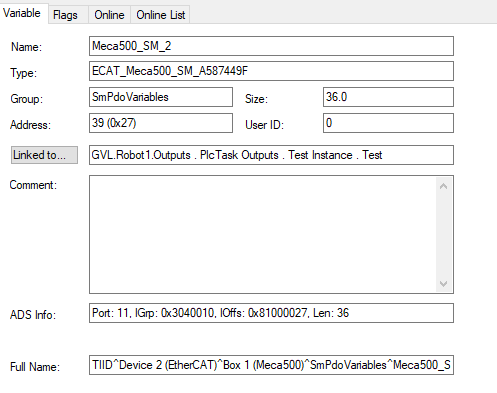
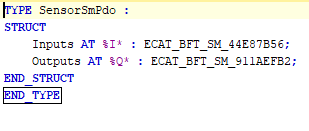
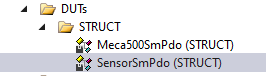
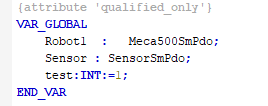
**Twin CAT PLC specific information**

In a PLC project there are different folders that can be used to organize your project



|  |  |  |
| --- | --- | --- |
| **Folder** | **Editability** | **Purpose** |
| External Types | No | Stores the custom objects of the project |
| References | Kind of (choose from options) | Stores libraries of the project |
| DUTs | Yes | Stores the structure of the custom objects |
| GVLs | Yes | Stores global variable list |
| POUs | Yes | Stores the scripts of code |

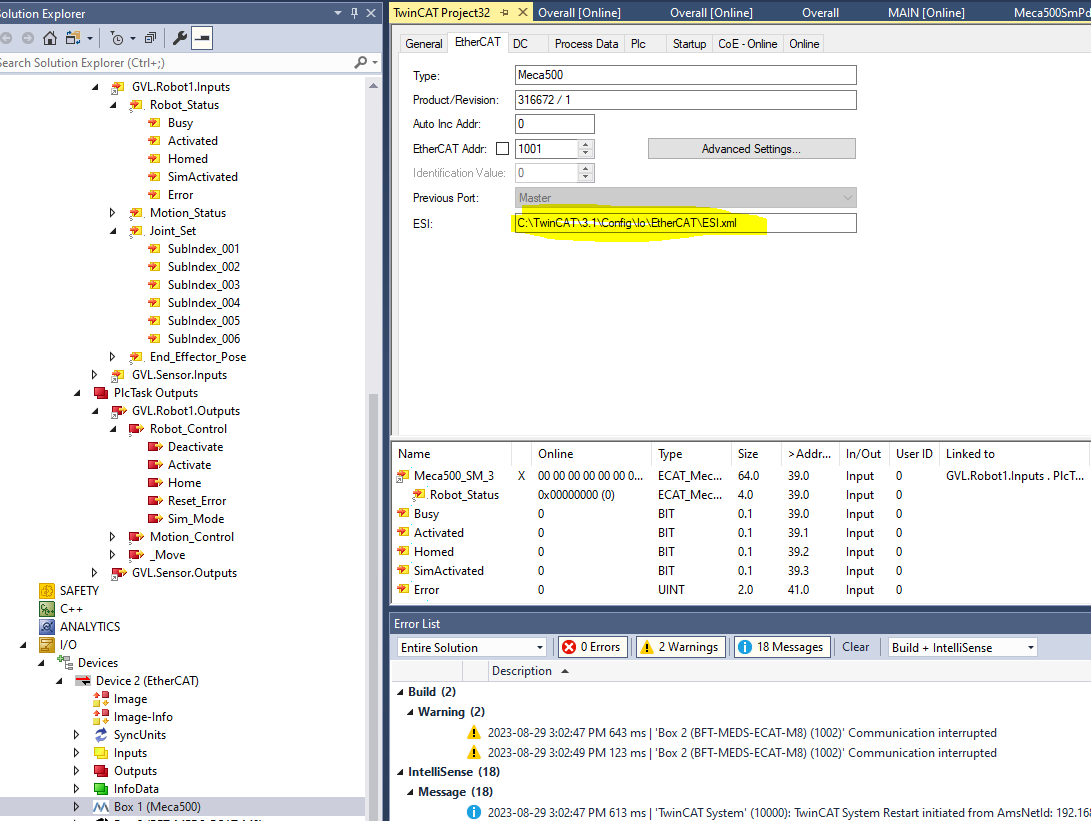
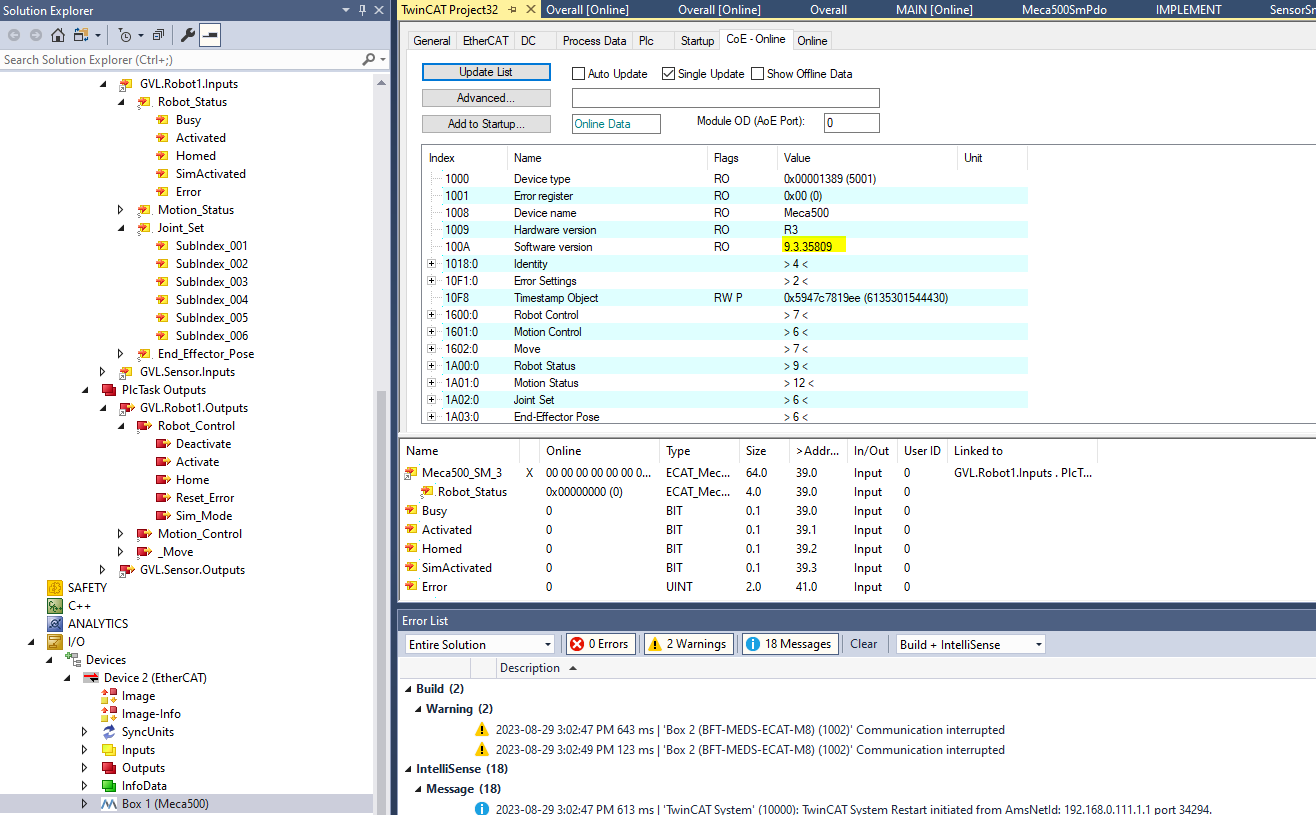
**What should be added to PLC project to use MECA and BOTA sensor data and communicate to**  **them**

* + **References**
    - \*\* add EtherCAT library
      * right click References folder --> Select add library --> Search for library using top search bar
      * 
    - References should look like image below
    - 
  + **DUTs**
    - \*\* add MECA and BOTA structures
      * Right click DUTs folder --> Select Add --> Select New Folder
      * Name folder STRUCT
      * Right click STRUCT folder --> Select Add --> Select DUT
      * Select “Structure” as type and Name DUT “Meca500SmPdo”
      * 
      * Copy info from Project32 into structure
      * 
      * Ensure input and output code match with that of the device. To check this follow steps below:
        + In IO/ Devices select the devices SmPdo Variables
        + 
        + The type should then match with that found on the devices SmPdo information, ie image below
        + 
        + Check this for both the input and output SmPdo and for all devices
      * This should be repeated for all devices connected through EtherCAT so that it can be used in the PLC project
        + Below is an image of the code for the BOTA sensor
        + 
        + If there are only a sensor and MECA500 connected through EtherCAT then the DUT folder should look like the below
        + 
  + **GVLs**
    - * \*\*Add global variable list
        + Right click GVLs folder --> select Add --> select Global Variable List
        + Add the global variables such as the one shown in image refer to Project32
        + 
  + **POU**
    - * \*\*Add main program and function that sends move data to robot
      * Refer to Project32 for specific program
  + \*\*After the above have been completed You can build the solution
    - At the top toolbar of TwinCAT click on Build --> Build solution

**How to run the PLC**

* When you have built the solution you can then run it
  + At the top toolbar click on the activate configuration button 
  + Then click on the login button 
  + Then click on the start button 
  + This will run your PLC
  + To stop press on the stop button 
  + To logout press on the logout button 

**OTHER INFO**

* ESI file should be downloaded into the folder found when you scan devices and should be named the same file name from the scan
* 
* Also make sure the ESI file matches the same Software version found in CoE-Online tab
* 
* After click on TwinCAT --> EtherCAT devices --> Reload Device description
* Close twinCAT and then reopen

**SOME TROUBLESHOOTING**

* If TwinCAT is unable to recognize the device or does not go into OP state
  + Go into the EtherCAT tab of the device
  + Record the ESI file location and name
  + Go into COE Online tab and note software version of the device
  + Search up the ESI files for the device and ensure it matches the software version
  + Place the ESI file into the appropriate file location and ensure name of ESI file matches
  + Reload device descriptions
* If TwinCAT is unable to run or if port cannot be found
  + Go into command prompt and run winsettick.bat