KLE Technological University
Creating Value
Leveraging Knowledge
DEPARTMENT OF AUTOMATION AND ROBOTICS

A Project Report

On

Automatic Storage and Retrieval System

Submitted in partial fulfilment of the requirement for the course of

INSTITUTIONAL SPONSORED PROJECT

21EARW492

Department of Automation and Robotics

Under the guidance of:

Arun C Giriyapur, Dr. Sachin Karadgi, Kartik Lakamanahalli.

.

Department of Automation and Robotics,

KLE Technological University, Hubballi. 580031

KLE Technological University
Creating Value
Leveraging Knowledge

DEPARTMENT OF AUTOMATION AND ROBOTICS

CERTIFICATE

Incorporated under KLE Technological University Act-2012; Karnataka Act 22 of 2013 ***Earlier known as***

B.V.Bhoomaraddi College of Engineering & Technology Vidyanagar, Hubballi

Department of Automation and Robotics

2021 - 2022

This is to certify that Mr. Amogh Sutar, Mr. Amrut Kurtkoti, Mr. Anant Maliye, Mr. K V S S Prathyush, Mr. Malhar Kulkarni, Mr. Prasad Basaragi, Mr. Ranjeet Shingade, Ms. Ritika Javalagi and has satisfactorily completed the course INSTITUTIONAL SPONSORED PROJECT sponsored by the KLE Technological University and LA Foundation for 6th and 7th semester in the University during the year 2021 -2022.

Staff in-charge **Head of the Department**

ABSTRACT

Warehouse management has been a key factor in ensuring that the company attains its complete efficiency and meets its productivity goals. The storing of the goods from the reception to the dispatch plays an important role in maintaining the shelf life of the goods and ensuring the proper dispatch of them as well. Due to the human errors and the mishaps by the labour involved in the warehouse, the automation of all the processes involved stands to be a more efficient and time-saving job. An Automated Storage and Retrieval System, shortly ASRS can substitute this in the warehouse management. Amazon warehouse, with its five-story plant, where the processes from the reception, storage, retrieving, and packing to dispatching have been automated, has been a key motivation to build this Automated storage and retrieval system.

Here, we report the designing, fabrication, and integration of an ASRS to store and retrieve the goods into the respective racks of the warehouse automatically. The proposed ASRS is a static serial robot that can move in all the 3 dimensions, with 3 DOF, and can pick or place i.e., store good in the rack or retrieve them from the rack, allowing it to move and function within the limits of its reach. The implications are, the ASRS should learn about the possible trends of goods regularly stored and retrieved for faster storing in the forthcoming cycles and also to reduce the cycle time for quick cycles. An Automated Storage and Retrieval System can account for all the goods, and maintain the warehouse by itself, substituting human effort and human error with accuracy and efficiency.

ACKNOWLEDGEMENT

We would like to take this opportunity to thank the following for the unique opportunity created through the institutional sponsored project to work on the given need statements.

Our sincere thanks to our professors Dr. Sachin Karadgi, Project manager, Mr. Kartik Lakamanahalli, and Mr. Arun C Giriyapur, Head of Dept. of Automation and Robotics, KLE Technological University (KLE Tech) for their valuable guidance, encouragement and suggestions throughout the project.

We would like to express deep gratitude to Mr. Arun C Giriyapur, Head of Department., Department of Automation and Robotics, KLE Tech for selecting us for the project.

Finally, we extend thanks to the entire faculty of the Department of Automation and Robotics, KLE Tech for their valuable support.

TABLE OF CONTENTS

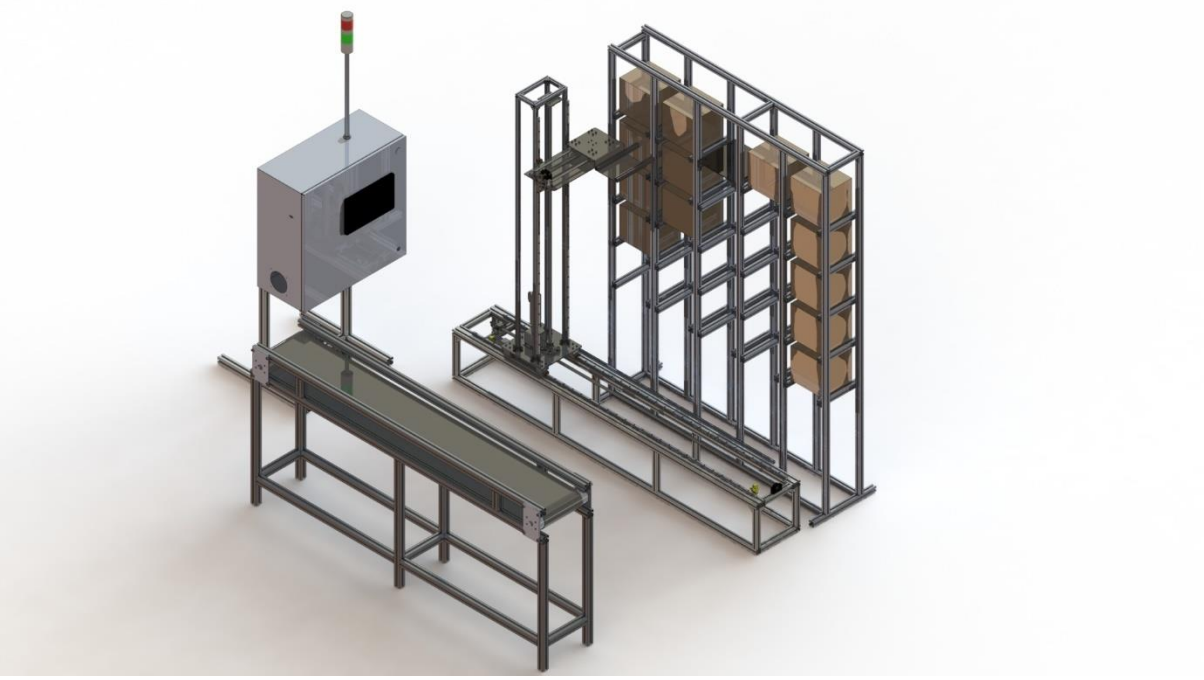
Chapter 1:

Introduction

Automatic storage and retrieval system (ASRS) consists of various computer-controlled systems to automatically place and unload loads at designated locations. Automatic storage and retrieval systems (ASRS) are commonly used in systems where:

- There is a very large amount of luggage loaded and unloaded
- Storage capacity is important due to space constraints
- There is no additional value for this process (no processing, storage and transport only)
- Accuracy is important because of the potential damage to the load.

The (ASRS) can be used with standard and non-standard loads, which means that each standard load can fit into a volume of the same size.



Automatic Storage and Retrieval System.

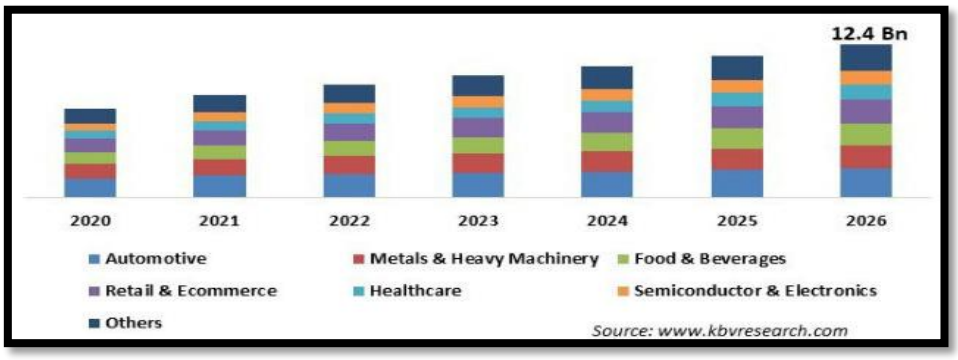
1.1. Problem Statement

Design a course on Digital Twin and development of automated storage and retrieval systems as an educational kit to complement the course on Digital Twin. Designing Digital Twin encompassing digital thread, monitoring and control of production processes and integrating the virtual world and physical world.

Chapter 2:

Literature Survey

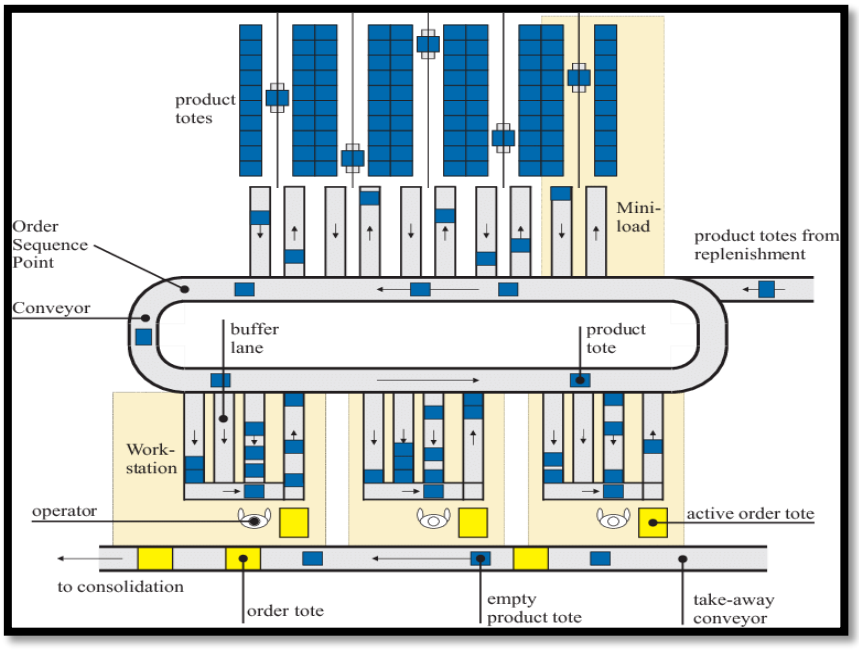
Automatic storage and retrieval systems have been widely used in distribution and production facilities since their inception in the 1950s. Between 1994 and 2004, there was a significant increase in the number of AS / RS used in distribution facilities in the United States (Production Phase for the American Automated Retail Industry Recovery System, 2005). The use of AS / RS has a few advantages over non-automated systems. Examples are savings on labour costs and floor space, increased reliability and reduced error rates.[1]



Automated Storage & Retrieval System (ASRS) Market Size 2026

Design and building a framework for automated equipment asset management system by reducing the lead time in the movement or development of thing and moreover to focus on improving Automatic Storage as well Retrieval System in the businesses to increase productivity and in addition improve asset management in the warehouse. The purpose is to plan and implement Automated Storage and Retrieval system of stockroom equipment and development, redesigning depends on parameters such as the point of the deal, the desired tone of the object or the visual parameters such as length, width, weight etc. using simplification procedures. Usage of SWARM optimisation algorithm can improve the overall performance of the system.[2] Particle swarm optimization (PSO) is a global optimization algorithm for dealing with problems where the best solution can be represented as a point or location in an n-dimension area.

Goods such as spare parts and small goods causes significant management problems. Because of the amount of space use and termination of filtering, this program is proposed handling goods properly by storing them in a specified location places safely and return them quickly without being damaged. A solid three-dimensional model of the proposed system is developed using computer-assisted drawing software. A program is written and a microcontroller is used to control the stepper motor for storage and retrieval of three dimensions. A basic idea about controlling scheme that is used in microcontrollers is obtained. [3]



Automated Warehouse Layout

An ASRS constitutes equipment and control which will handle the storage and retrieval of required material with great accuracy and precision. There are different variety of ASRS systems that vary from small automated systems to large automated systems which are usually an integral part of distribution and manufacturing process.

They are widely used in the logistics as well. Rising trend of e-commerce has demanded these logistics and warehouse companies to work faster in order to satisfy the customers and keep up the company's reputation. In order to keep up with the orders these firms also follow some key performance indicators (KPIs). Some of the KPIs are Average Hourly Usage, Average Weekly Usage, Shortest Path, Cycle Time, and Accuracy. Etc.

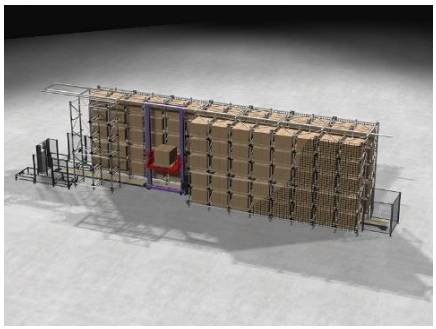
Based on these KPIs we can analyse the performance of the machine and we can carry out predictive maintenance activity so that there is no wearing away of machine parts.[4]

Depending on the load, we can classify them into two types of AS/RS machines:

1. **Mini-Load AS/RS Machine**: This machine is meant for small load usually which weighs less than 1000 pounds. They are typically for lighter loads.

Mini Load AS/RS Machine

2. **Unit-Load AS/RS Machine**: This machine is typically is meant for heavy loads which usually weighs more than 1000 pounds and more. The pallet and the whole structure will be as tall as 100 feet or even more than that.[5]



Unit Load AS/RS Machine

Chapter 3:

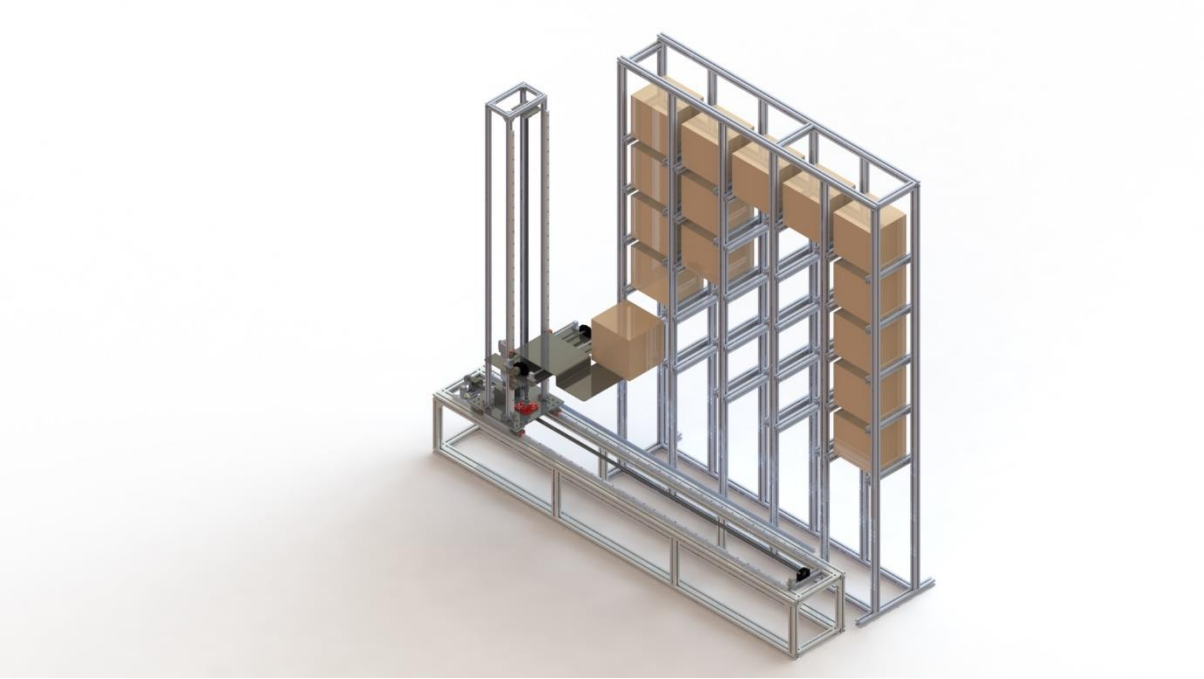
Design, Fabrication and Architecture

3.1. Design

An abstract idea for the design of an Automated Storage and Retrieval system would be the combination of equipment and controls which handles, stores, and retrieves materials with precision, accuracy and speed under a defined degree of automation and a rugged industrial environment [6].

3.1.1. Mechanical

The mechanical task included for the designing phase of our Automated Storage and Retrieval system deals mainly on the design and fabrication and analysis of the materials used for the assembly [7] [9]. The complete mechanical work followed here is based on the V- model for manufacturing of products [8]. The ASRS designing and assembly is followed by parts, it is divided among X-axis, Y-axis and Z-axis and rack assemblies.

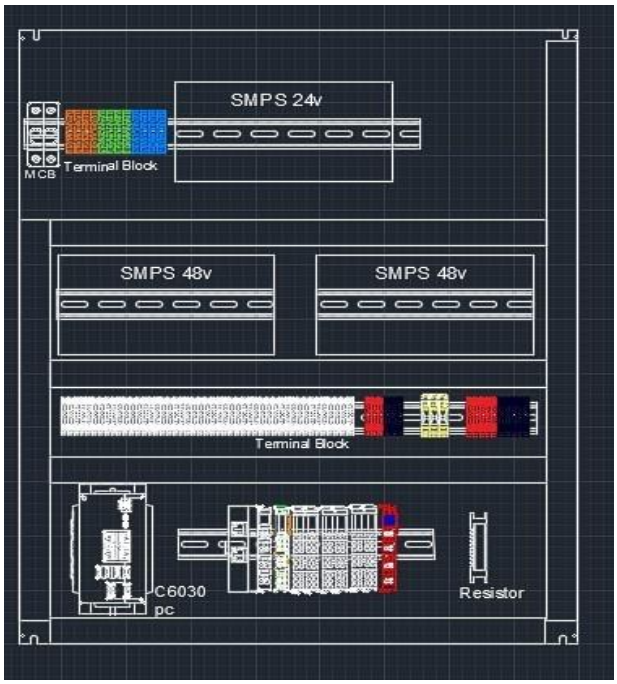


Rendered Image of the Solidworks Design of ASRS

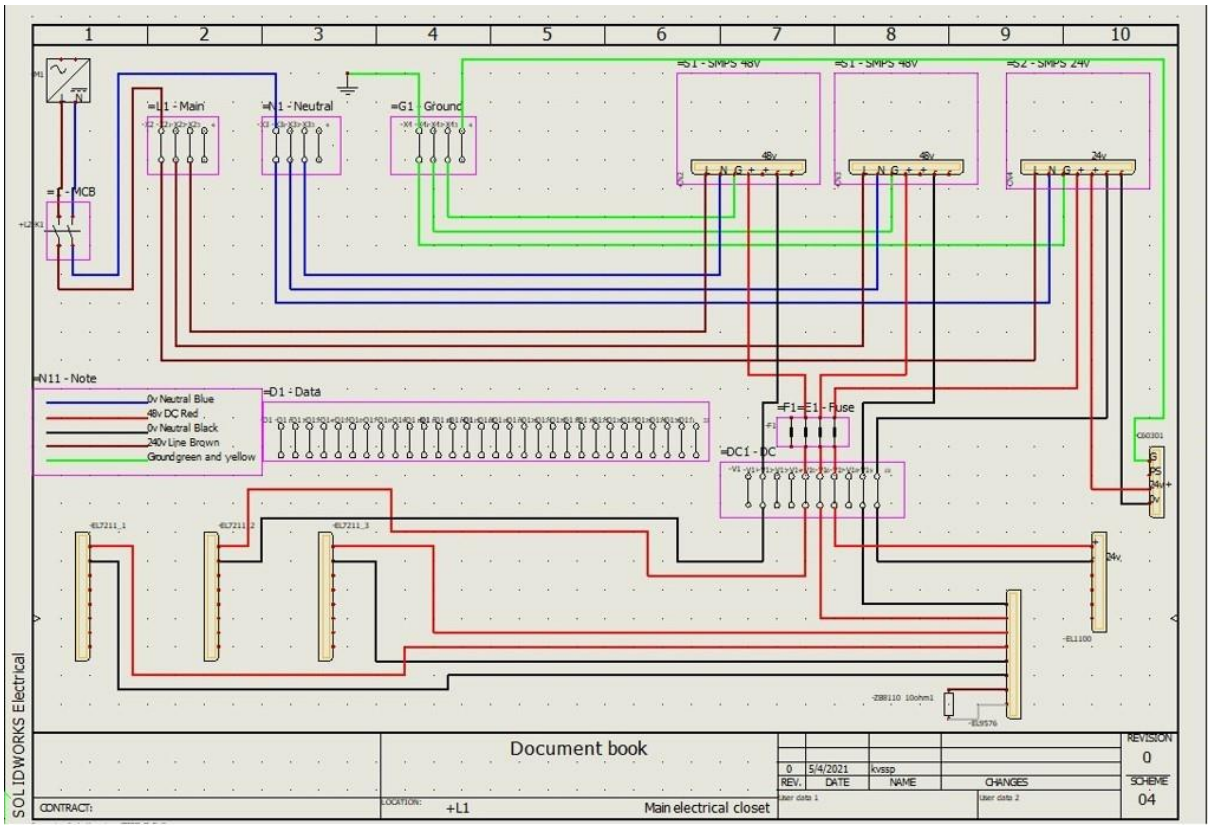
3.1.2. Electrical

The placement of electric components has been made in AutoCAD software following IEEE standard of industrial guidelines with separating AC (220V) wiring to left and DC (48v) wires to the right. Wiring regulations often define a maximum fuse current rating of (10A) for these circuits. Overcurrent protection devices are essential in electrical systems to limit property damage and threats to human life. This standard provides test criteria to determine the suitability of heating devices and fittings that are used for commercial applications. The standard also includes detailed recommendations for the design, installation, and maintenance of electrical resistance heat tracing in these applications.

AutoCAD and Solidworks Electrical Wiring Diagrams



Control Panel Design and component placement.



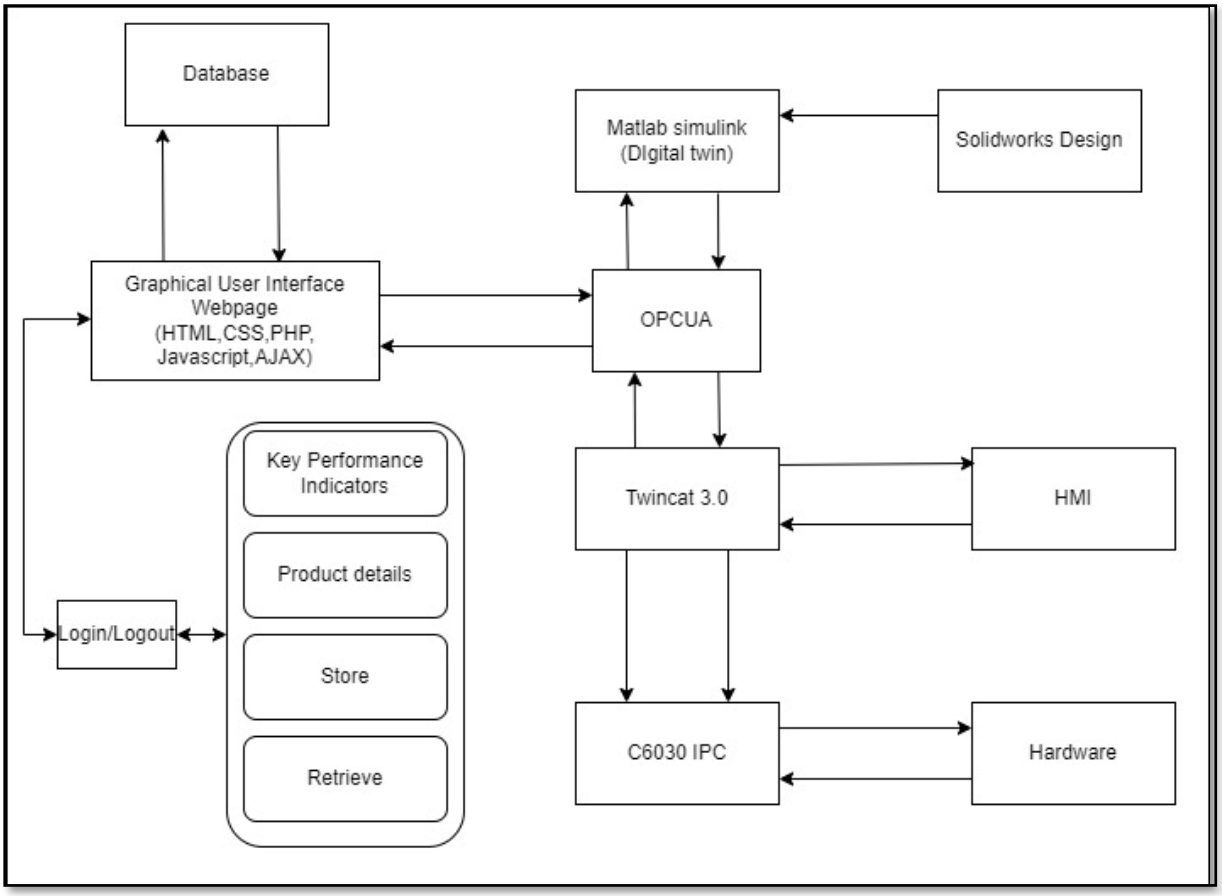
Wire diagram of electrical components

3.2. Fabrication

Fabrication of our approach to the problem statement had many challenges as well as learnings for developing a functional and rugged Automated Storage and Retrieval system. Fabricating of the designed solution involved many machine operations such as Cutting of Aluminum beams, Laser Cutting of Sheet Metals, grindings, deburring, bore drill, Tapping operations, 3D Printing, Fastener Application, etc.

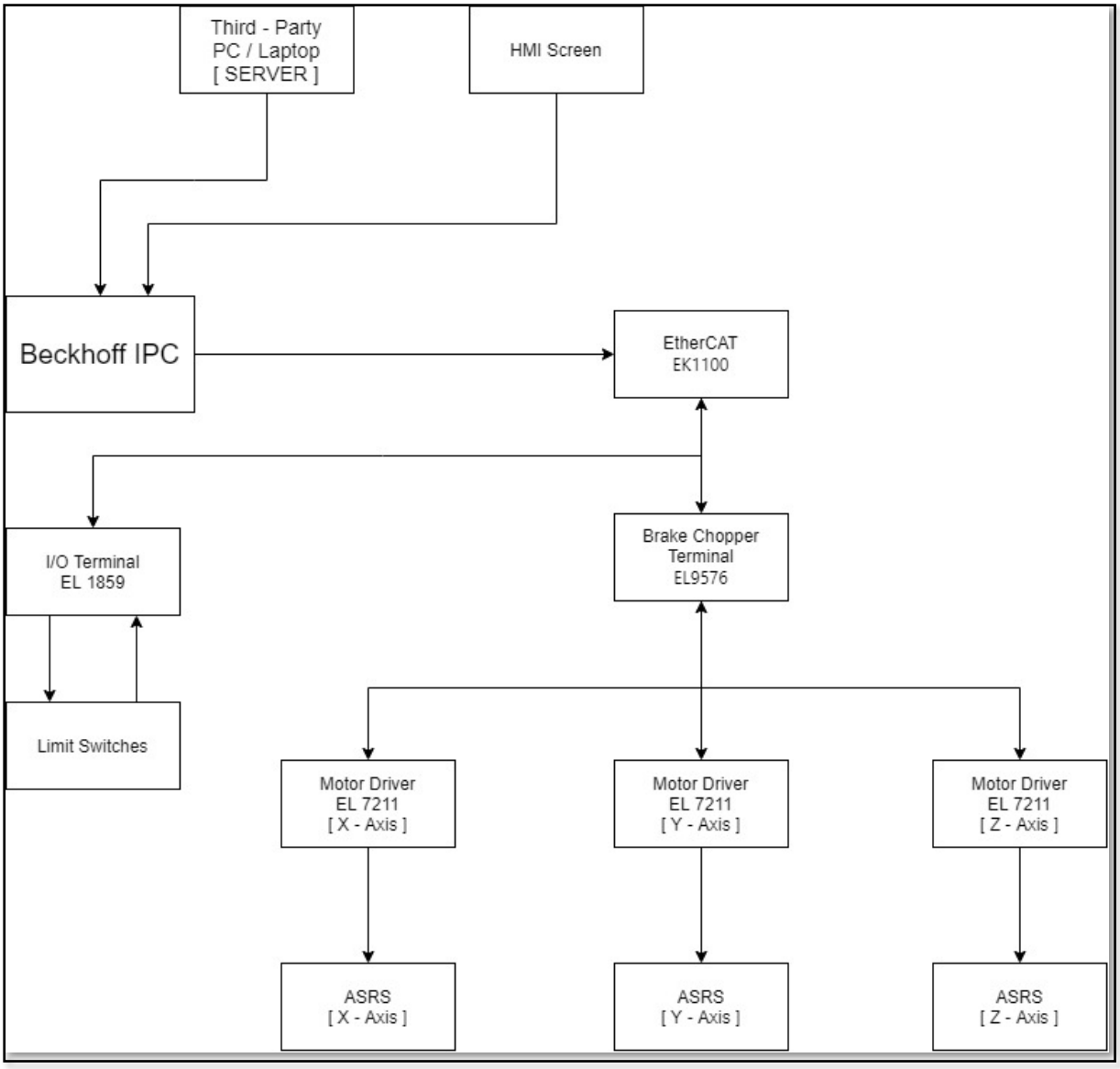
These are some of the processes that are implied for fabricating and assembling various parts for the machine. The operations that are carried are done under the guidance of machine operation experts and proper safety measures are taken before the operations.

3.3. Software Architecture

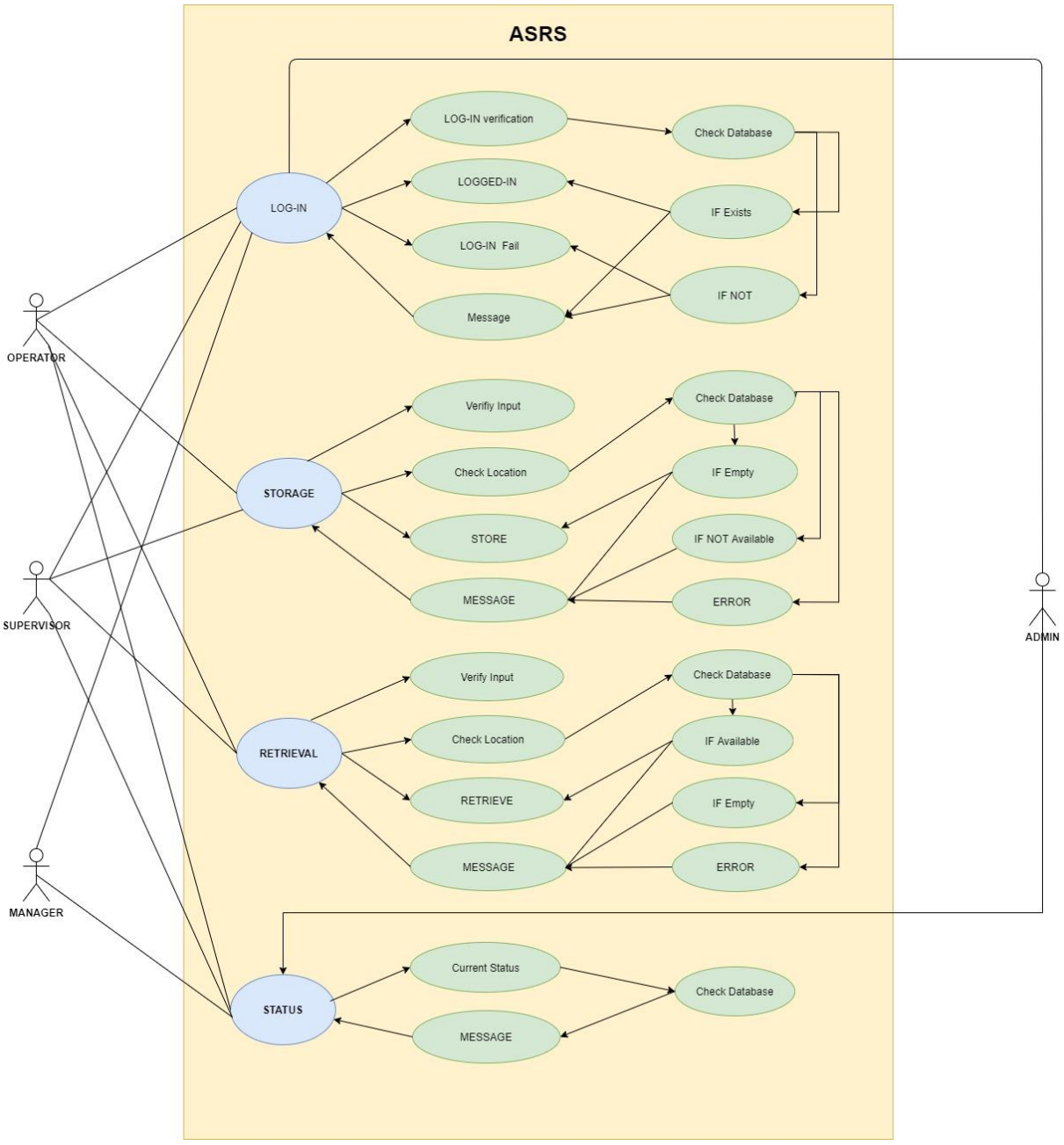


An automated storage and retrieval system (ASRS) is a high-rise system which consists of computer-controlled methods for automatically placing and retrieving loads from specific storage locations. Thus, reduces human intervention and avoids human error. C6030 Beckhoff IPC is used to control the hardware system with the structured text programming done with Beckhoff Twincat 3.0 software, it is used to configure and control motors. We have created a HMI to control the whole ASRS for storage and retrieval mechanisms. OPCUA network is being used to access and control the system for far distance and it is also used for digital twin under local ethernet. A Solidworks model is been imported to MATLAB-Simulink and code is been written to access the variables from TwinCAT to MATLAB for digital representation and controlling of ASRS.OPCUA network is been linked to the webpage interface where we can monitor the Key performance indicators and product details and can control the storage and retrieval operations from this webpage interface.

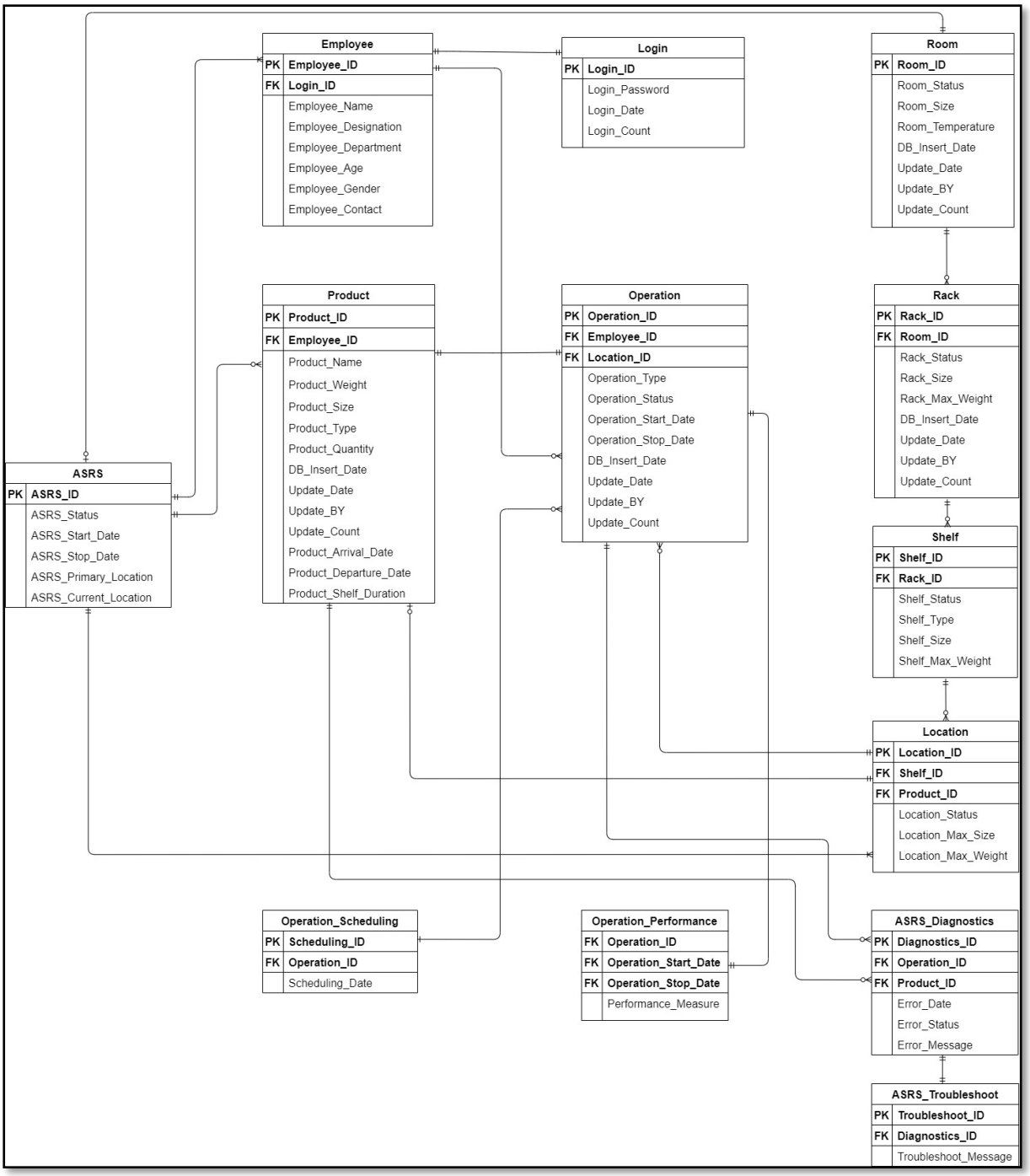
3.4. Hardware Architecture



3.5. Use Case Diagram



3.6. ER – diagram



Chapter 4:

Methodology

To solve any engineering related problems, we need to follow certain procedures or methodology which further helps in better understanding of the problem and leads to solution which can be modified and updated according to different needs and purposes.

The best methodologies that solve any engineering related problem with whole new perspectives are Agile Methodology and Engineering Design.

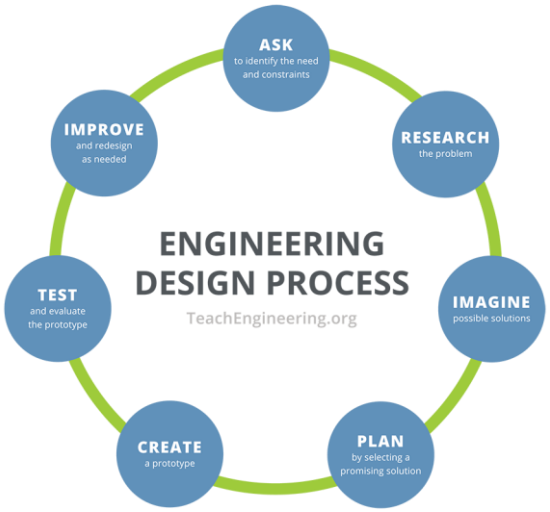
Agile Methodology is a process which promotes continuous iteration of development and testing throughout the software development lifecycle of the project. The development of the model and testing of the model goes hand in hand in this methodology.

The Agile Methodology follows the following steps

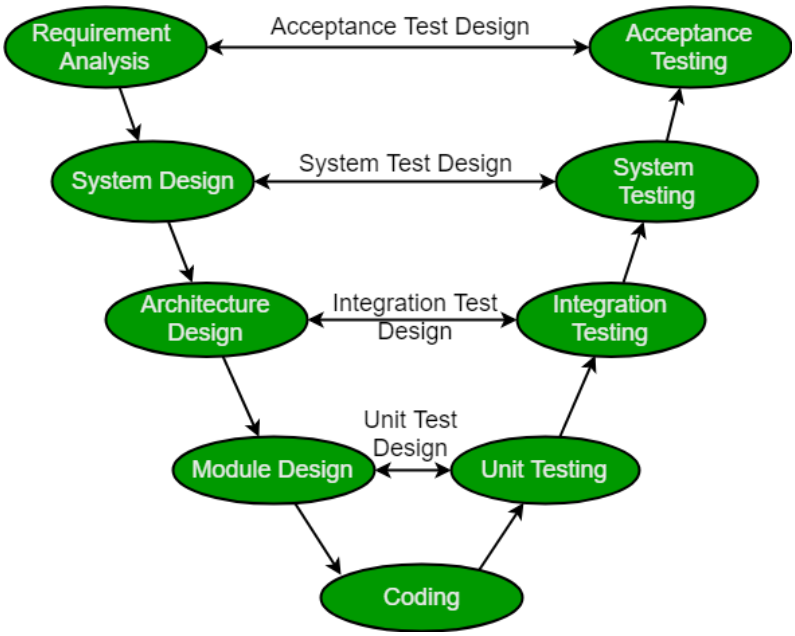
1. Requirement's definition and Analysis of concepts
2. Planning of sprints
3. Collaborative design development
4. Creation and implementation
5. Review and Monitor.

Source: https://www.nvisia.com/insights/agile-methodology

The engineering design process (ED) is a process which has series of steps that engineers follow to come up with a solution to any given need statement or problem. This process is followed when the product involves designing, building and testing.



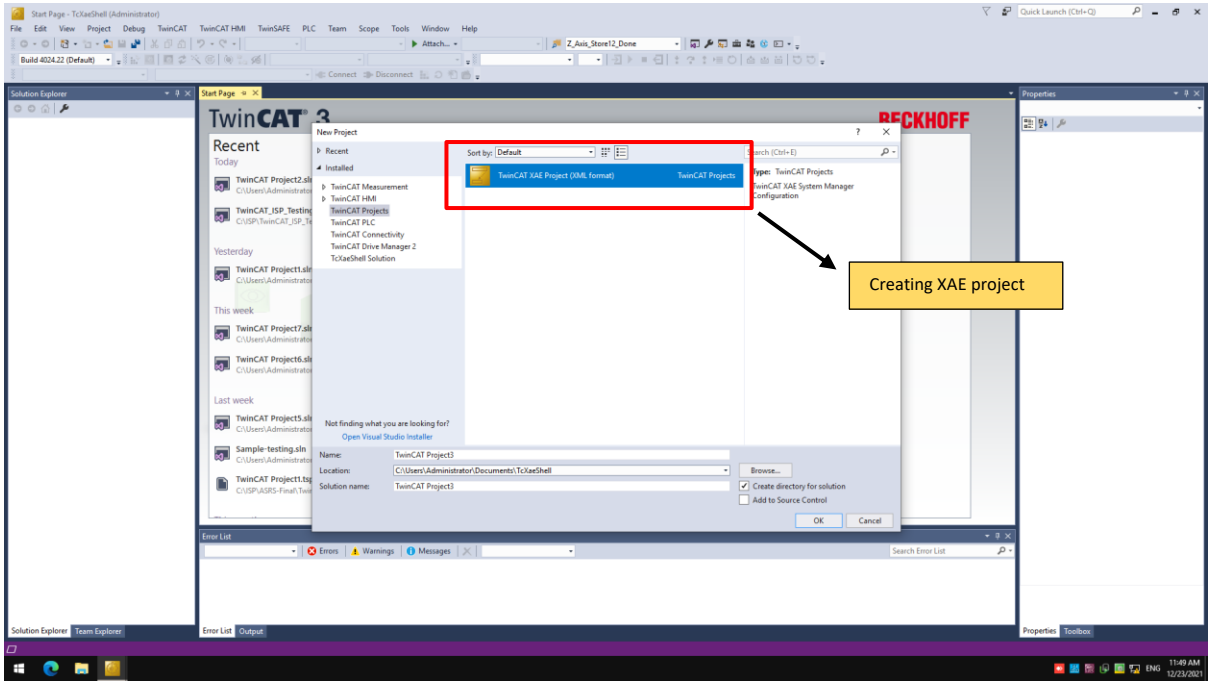
The V-model is verification and validation model. It is based on the association of a testing phase for each corresponding development stage. Development of each step directly associated with the testing phase. The next phase starts only after completion of the previous phase i.e. for each development activity, there is a testing activity corresponding to it.

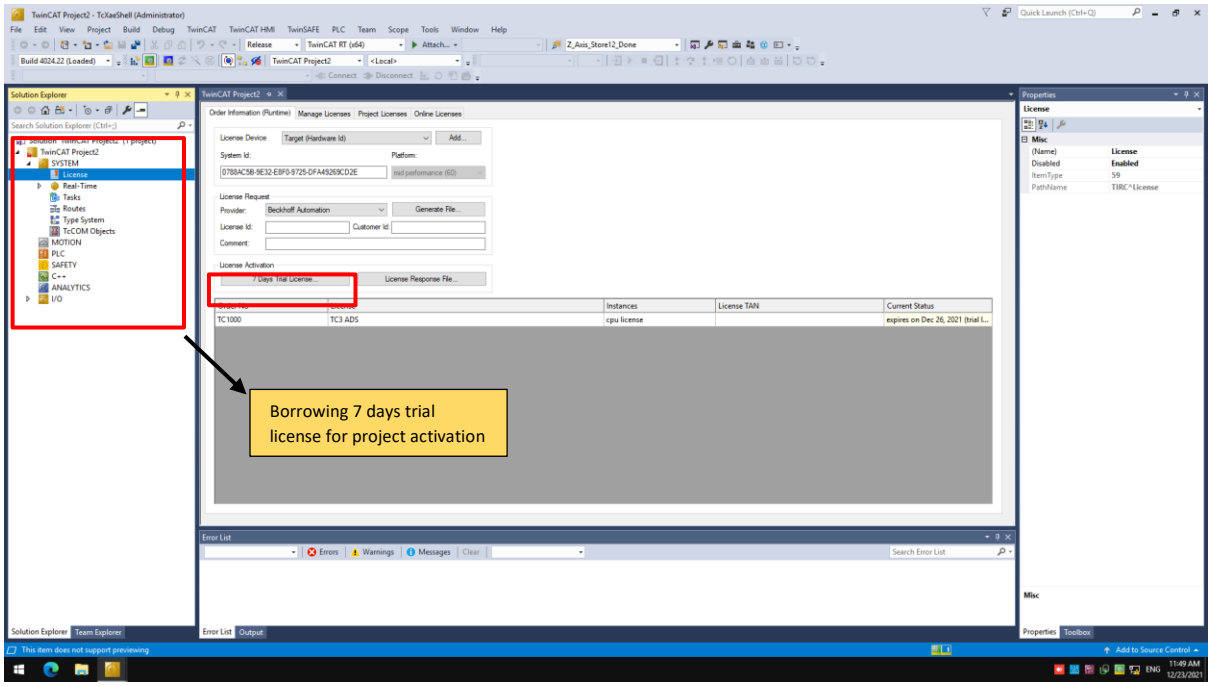


4.1. TwinCAT

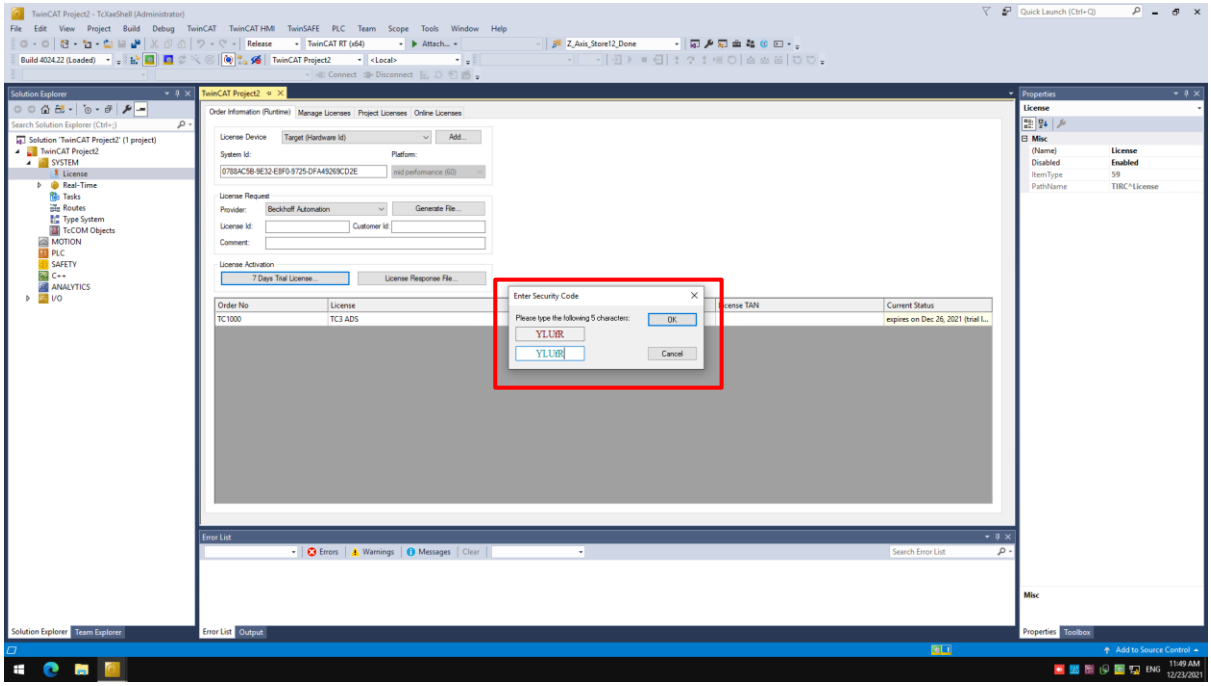
The TwinCAT (The Windows Control and Automation Technology) automation suite forms the core of the control system. The TwinCAT software system turns almost any PC-based system into a real-time control with multiple PLC, NC, CNC and/or robotics runtime systems.

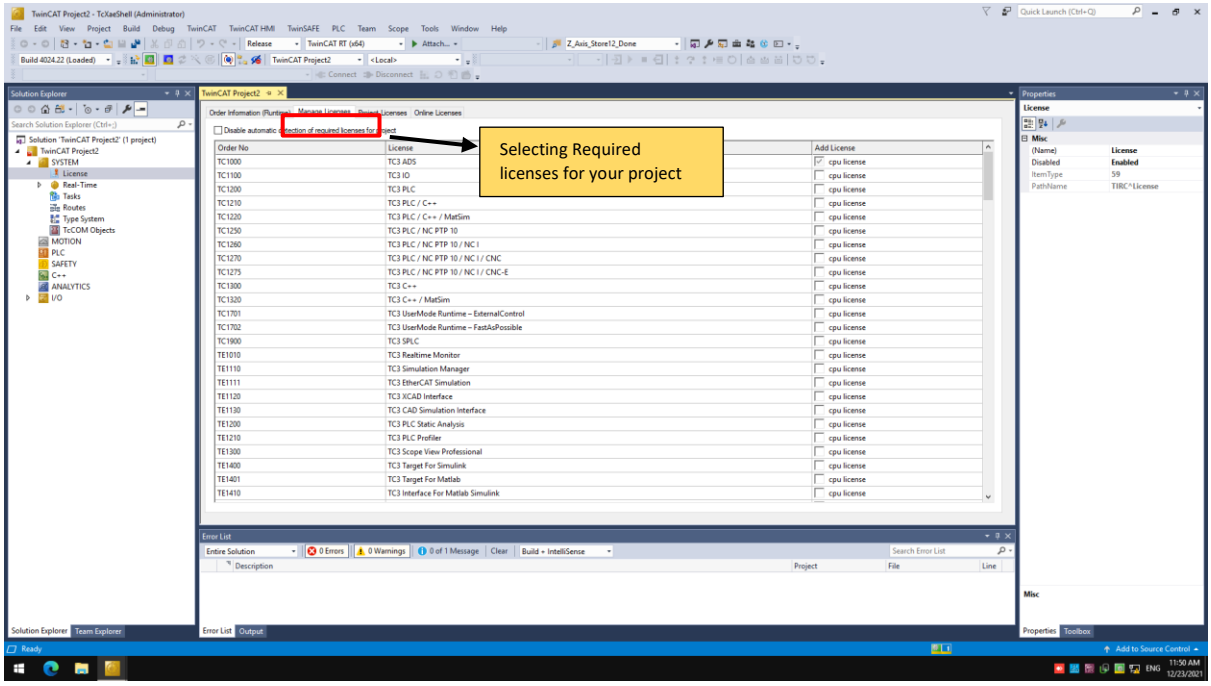
4.1.1 Creating Twincat Project



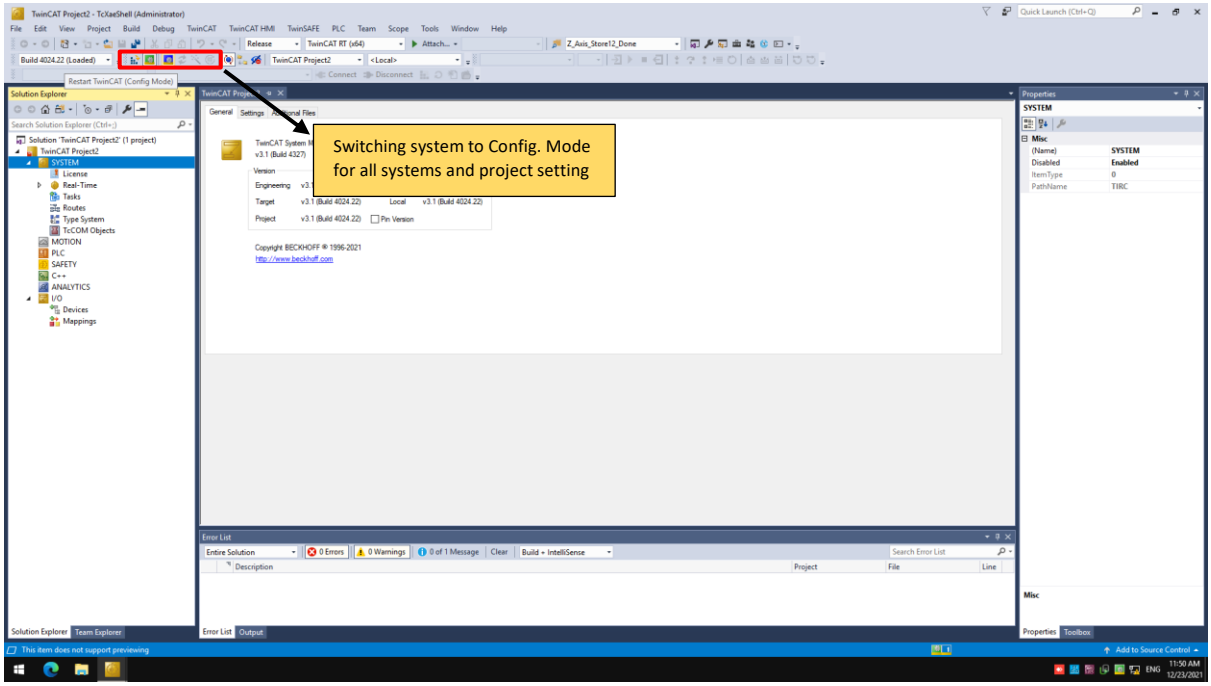


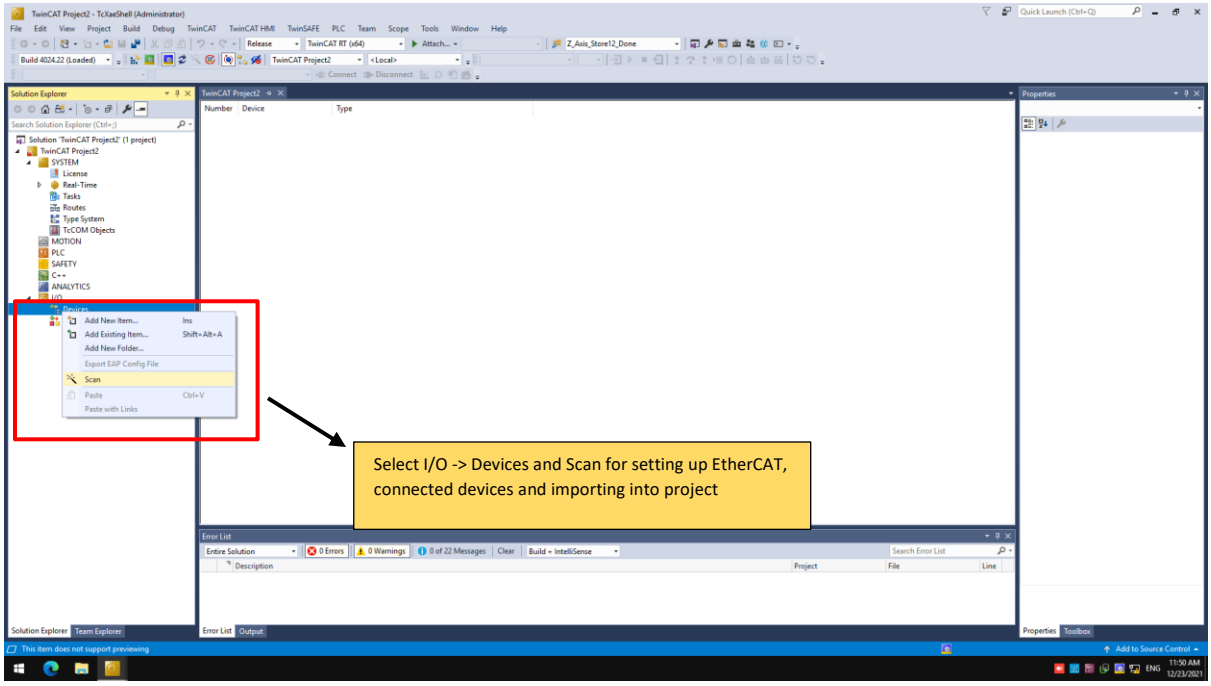
4.1.2 Borrowing License for Project

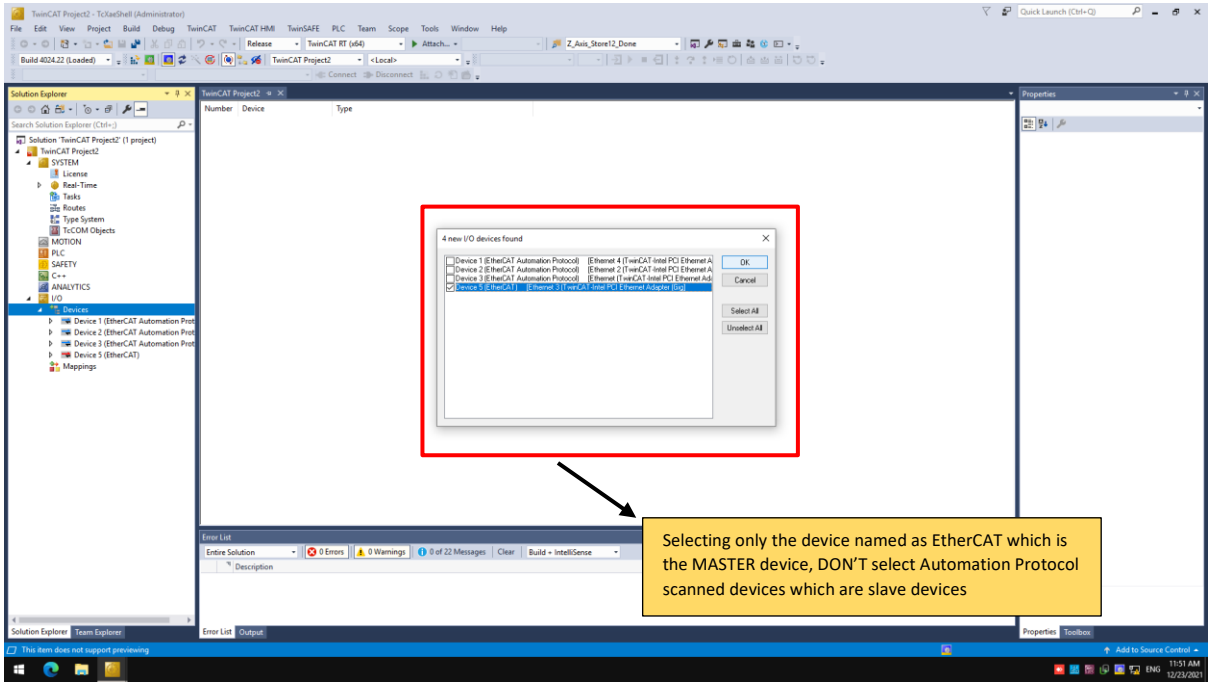


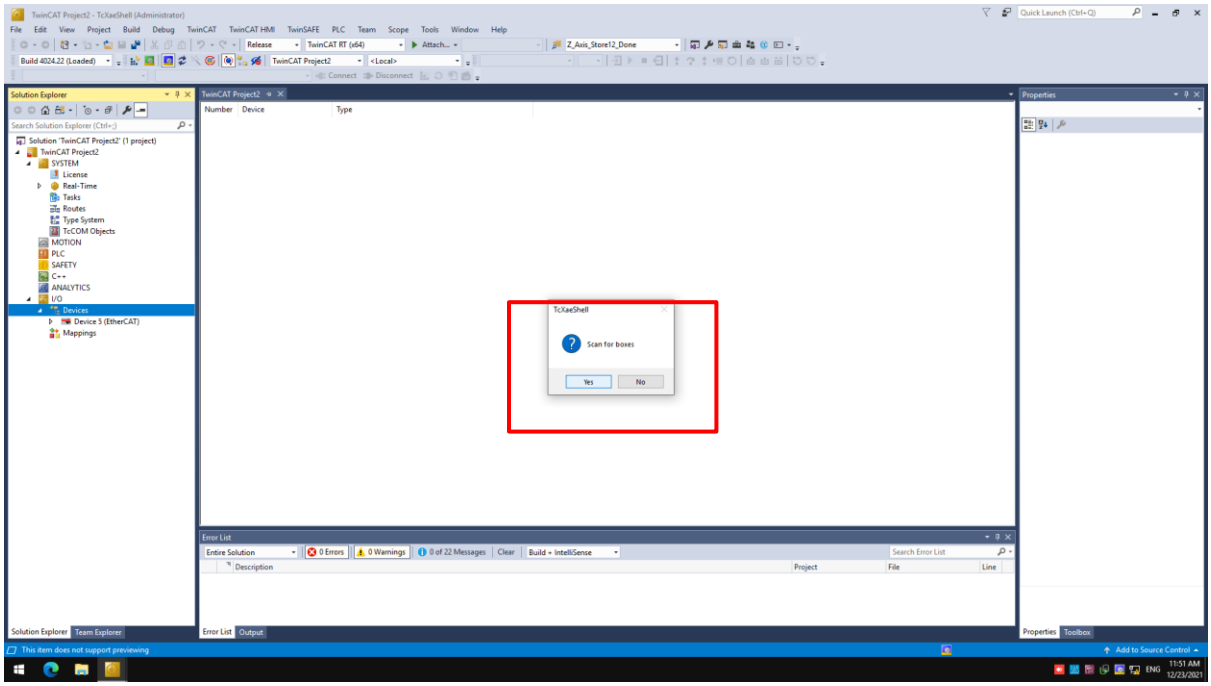


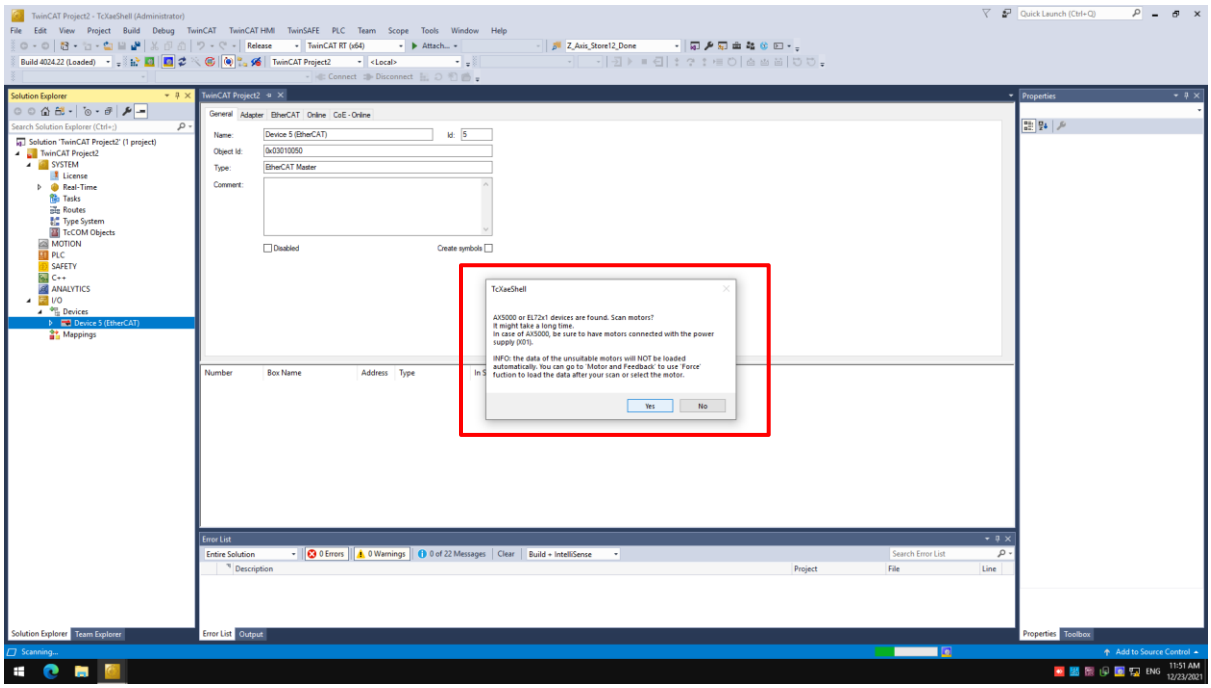
4.1.3 Initial Project Setup and Scanning of Beckhoff Hardware

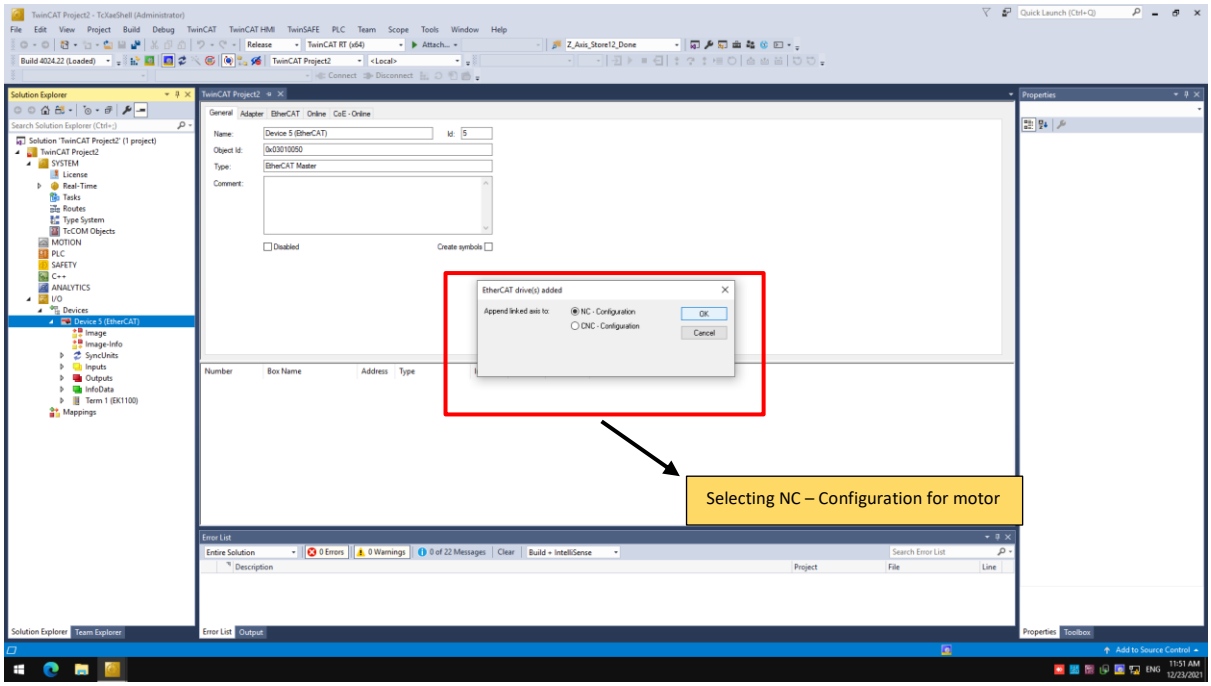


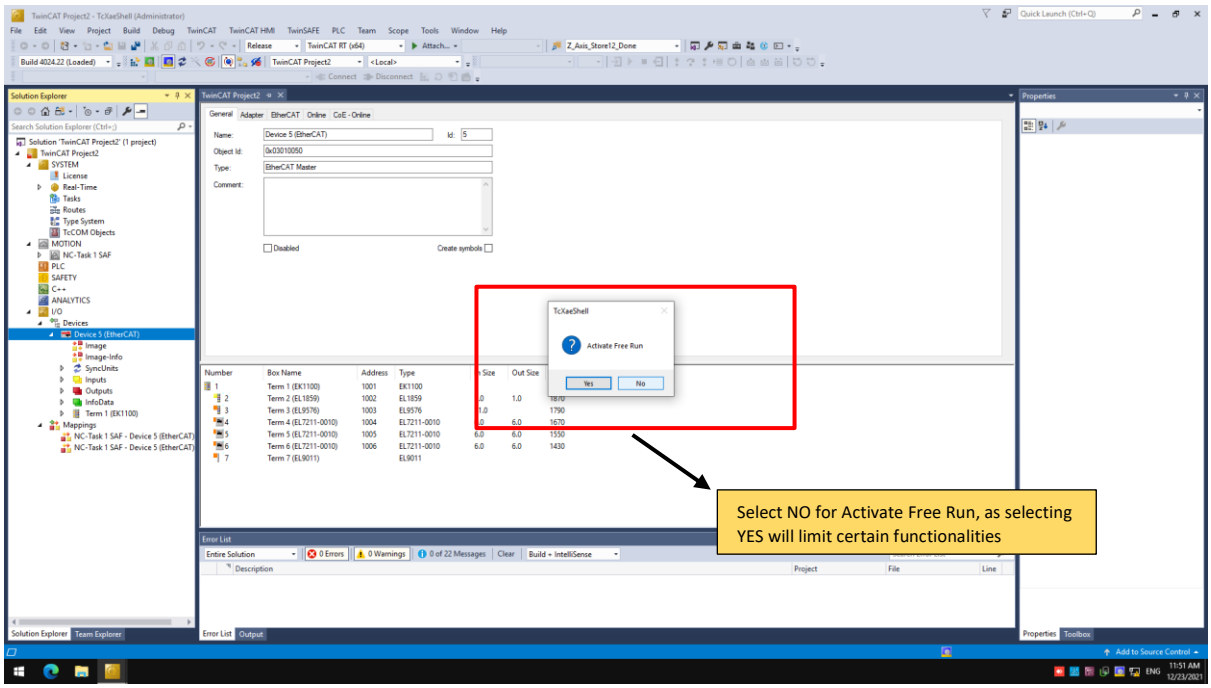


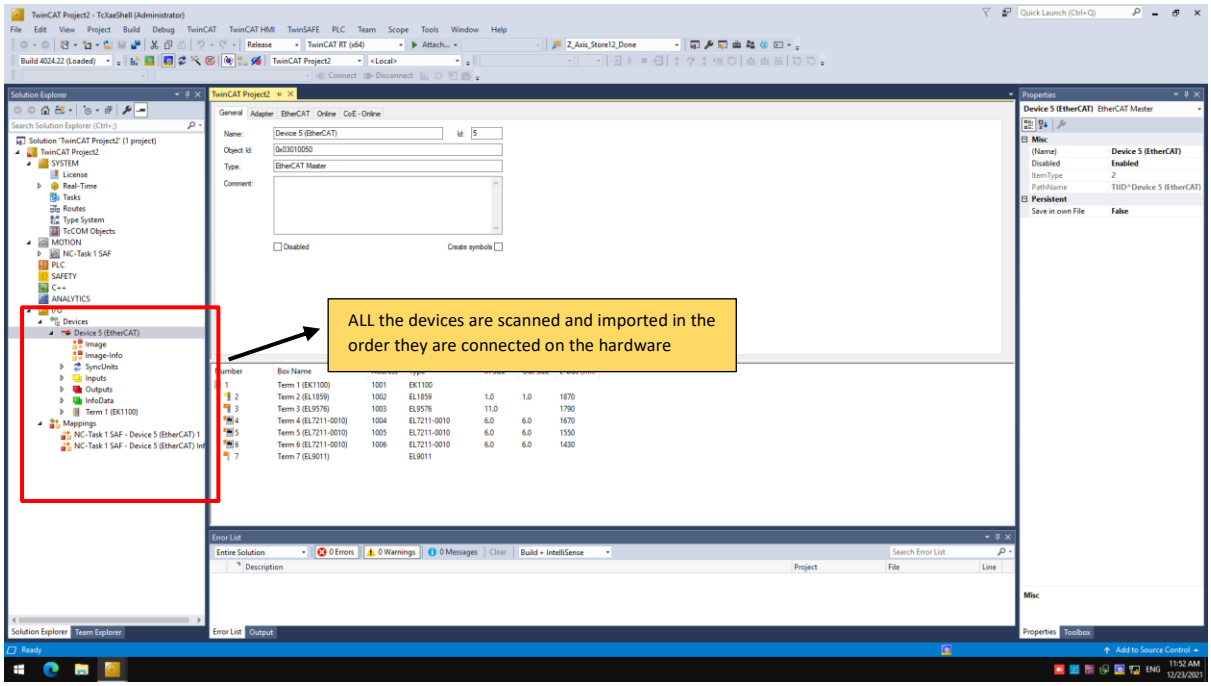

4 new I/O devices found
OK
Cancel
Select All
Unselect All
Selecting only the device named as EtherCAT which is the MASTER device, DON'T select Automation Protocol scanned devices which are slave devices


TcXaeShell
Scan for boxes
Yes
No

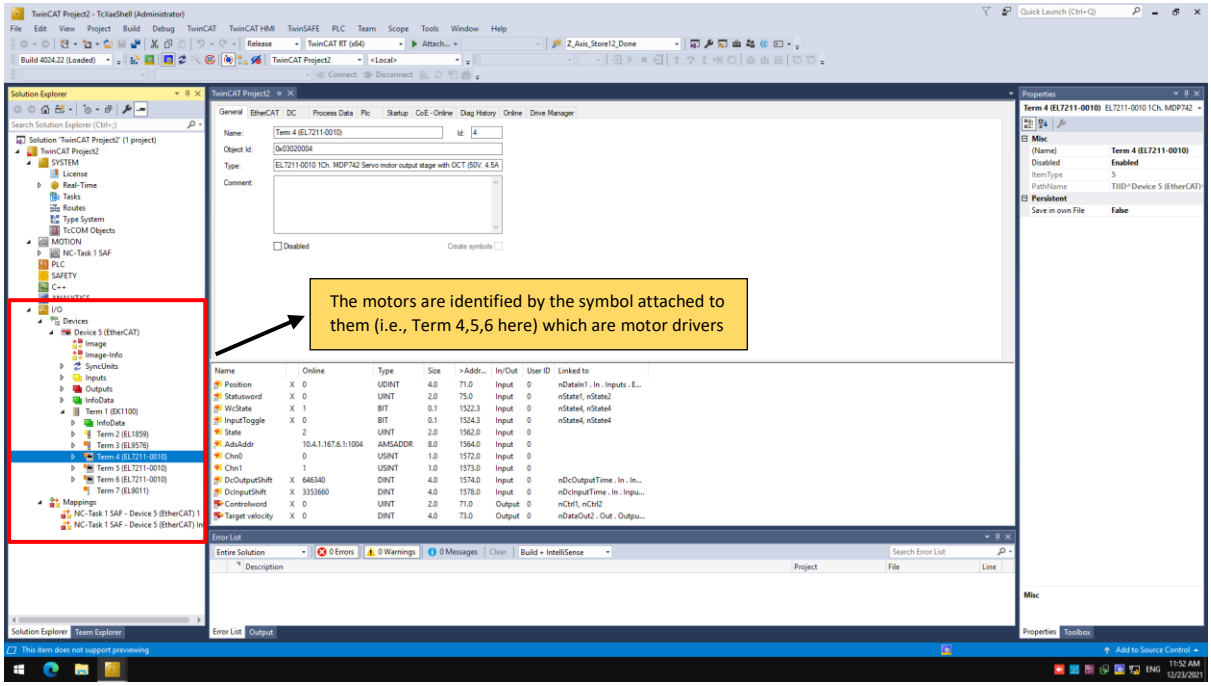

TcXaeShell
AX5000 or EL72x1 devices are found. Scan motors?
It might take a long time.
In case of AX5000, be sure to have motors connected with the power supply (K01).
INFO: the data of the unsuitable motors will NOT be loaded automatically. You can go to 'Motor and Feedback' to use 'Force' fuction to load the data after your scan or select the motor.
Yes
No

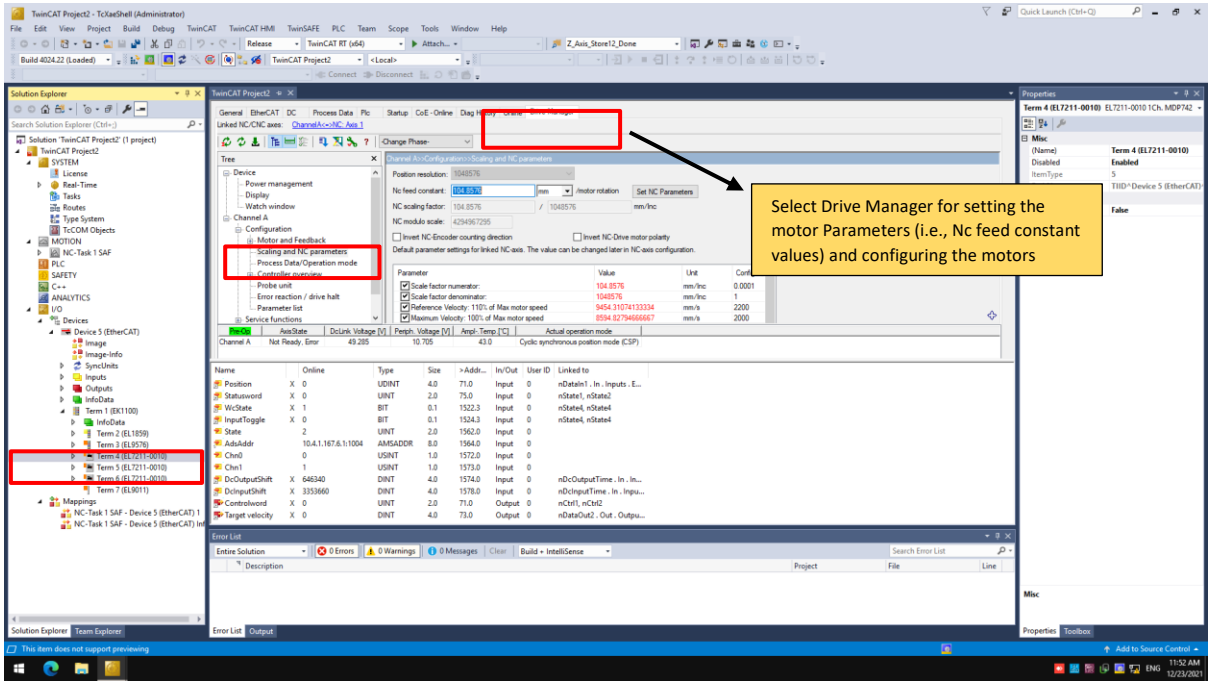

EtherCAT drive(s) added
Append linked axis to:
NC - Configuration
CNC - Configuration
OK
Cancel
Selecting NC – Configuration for motor

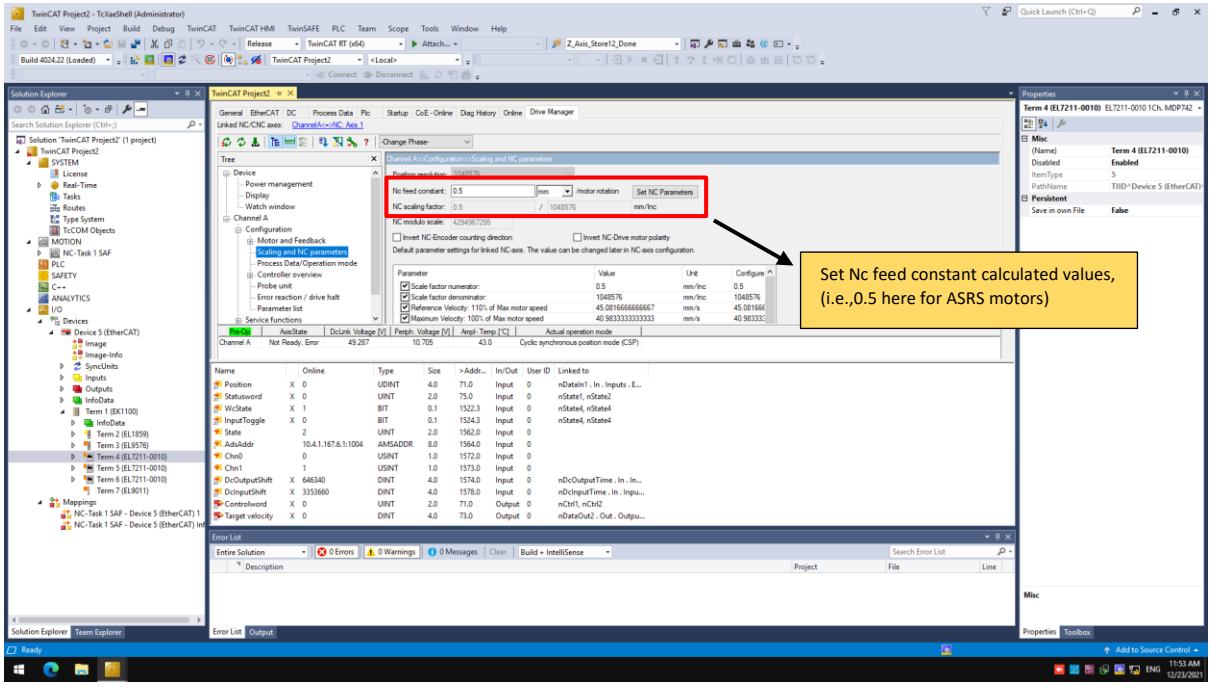

Select NO for Activate Free Run, as selecting YES will limit certain functionalities

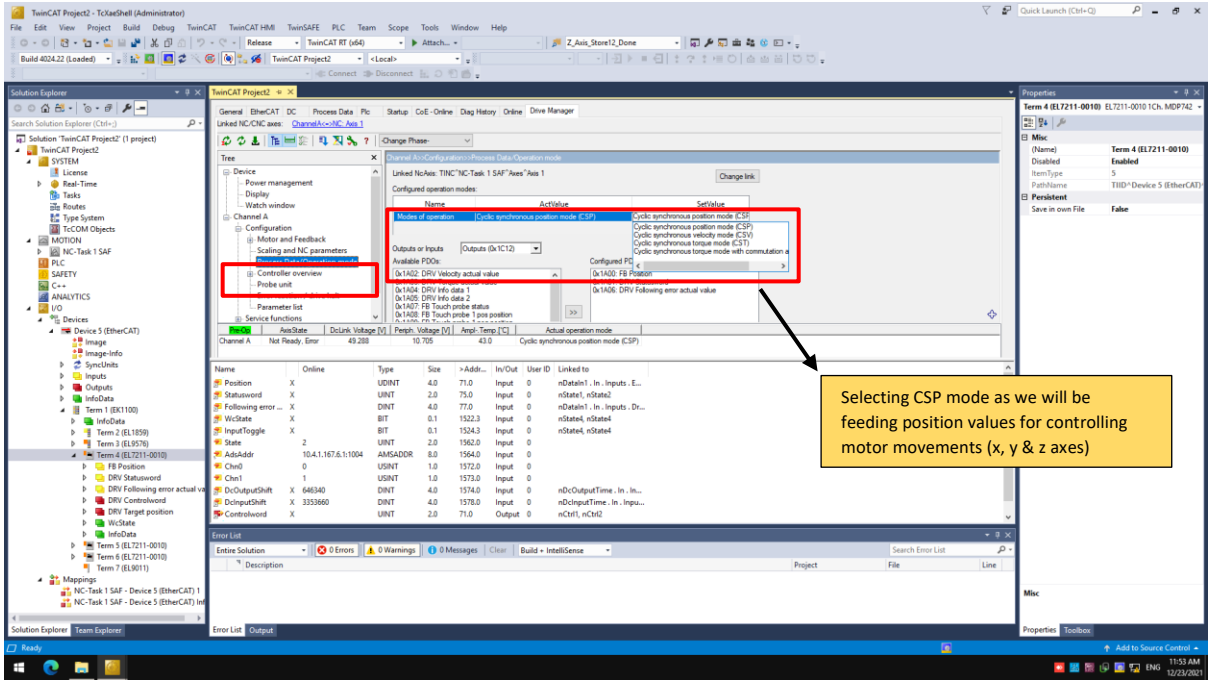

ALL the devices are scanned and imported in the order they are connected on the hardware

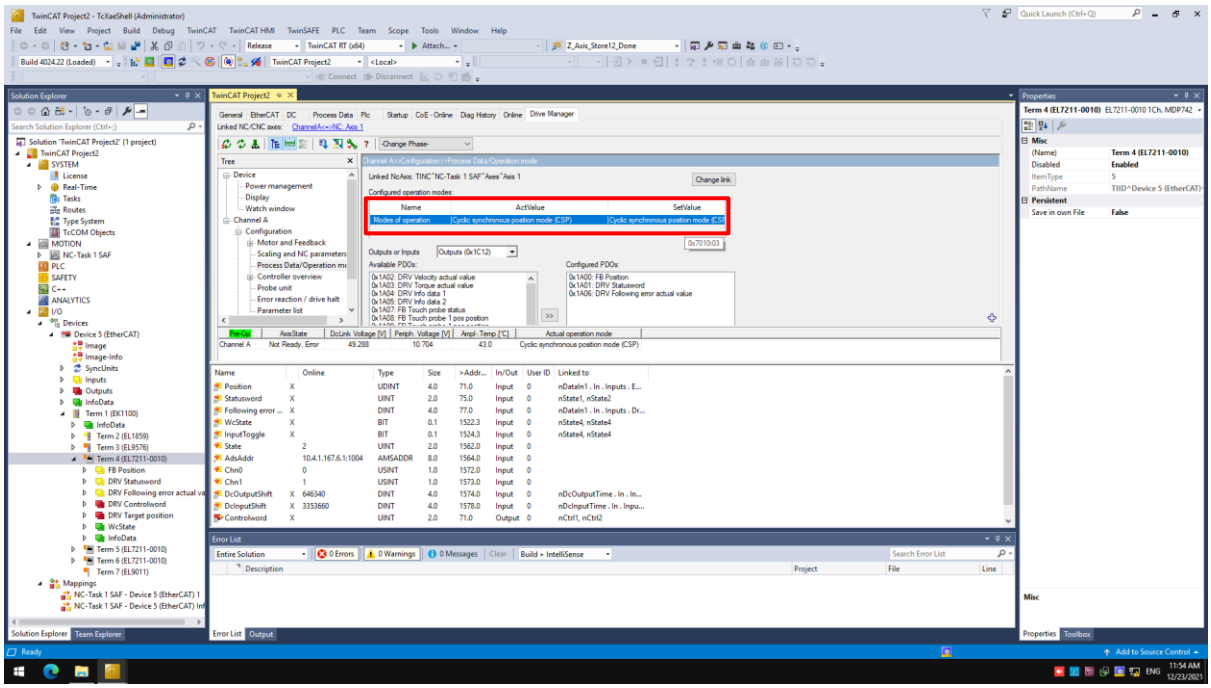
4.1.4 Motor Parameters Configuration

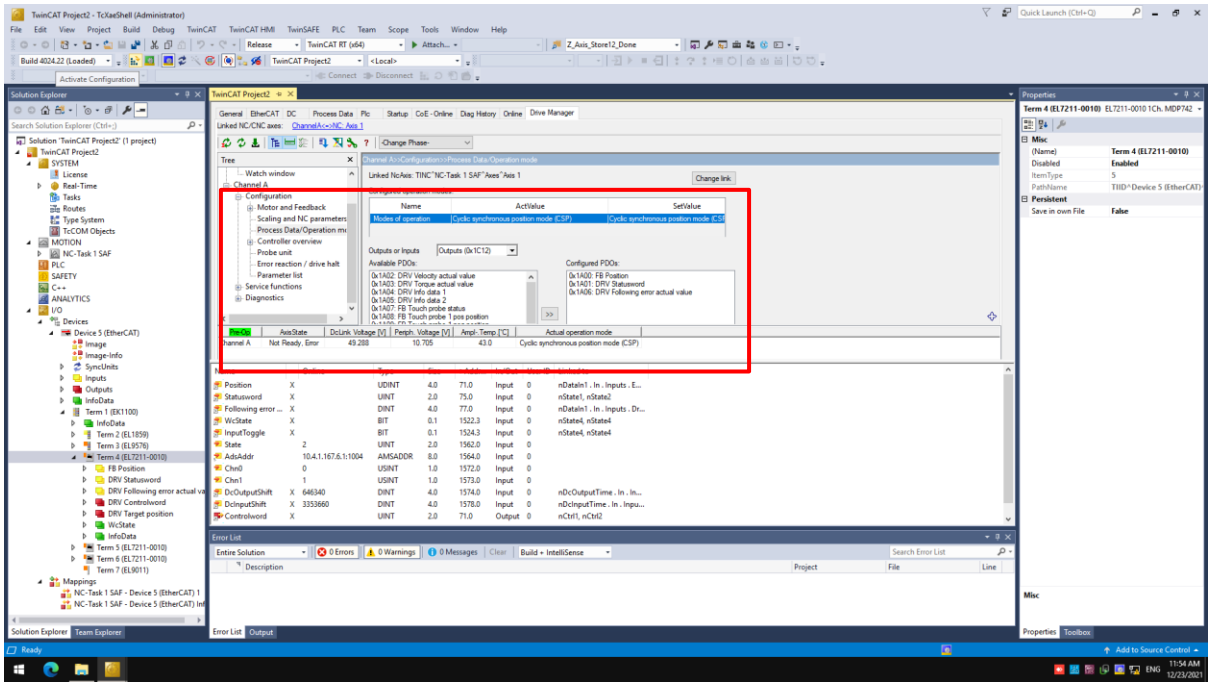




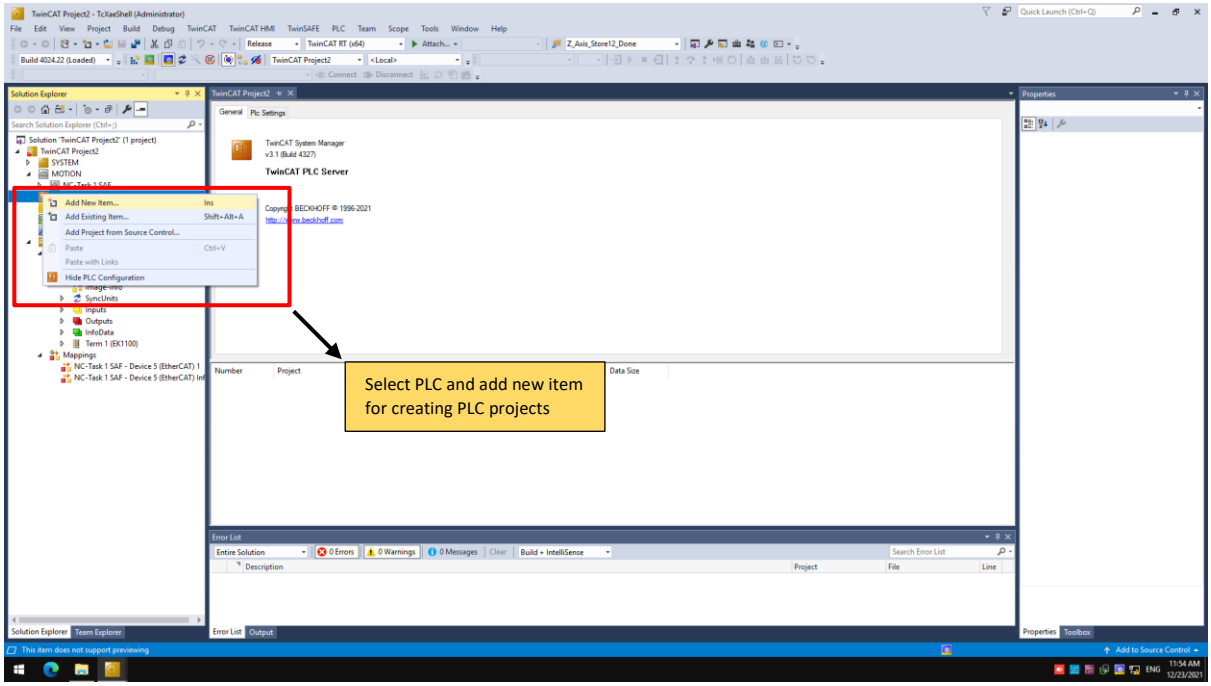

Set Nc feed constant calculated values, (i.e.,0.5 here for ASRS motors)

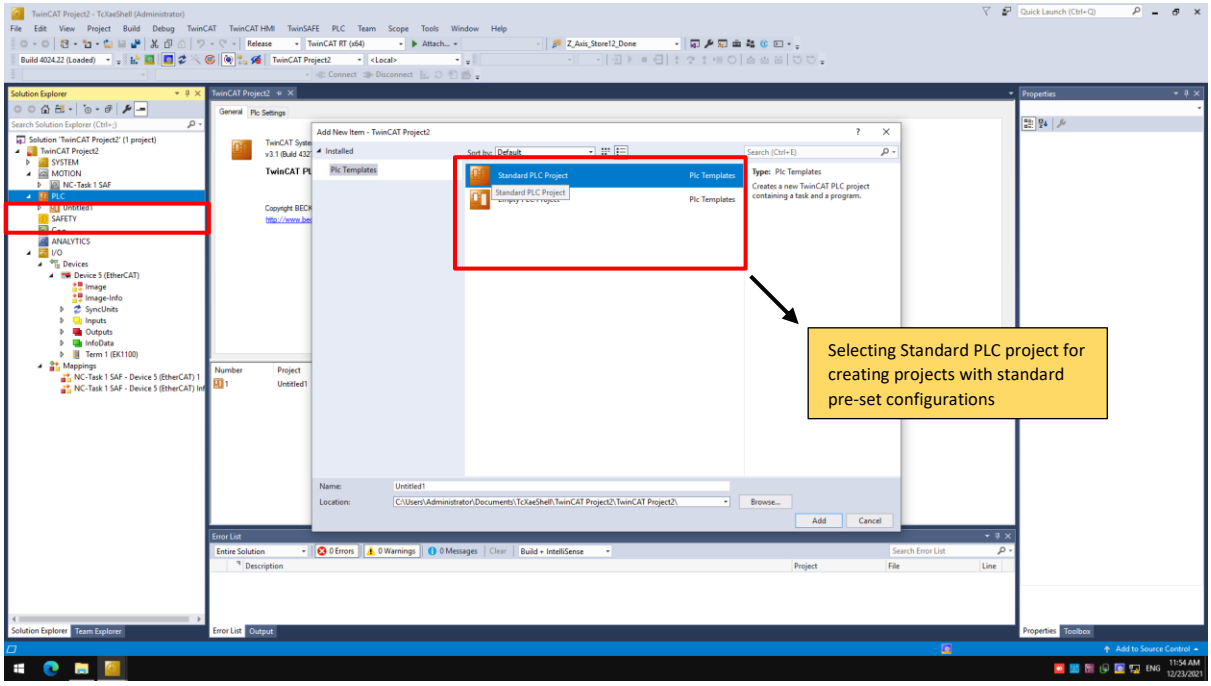

Selecting CSP mode as we will be feeding position values for controlling motor movements (x, y & z axes)

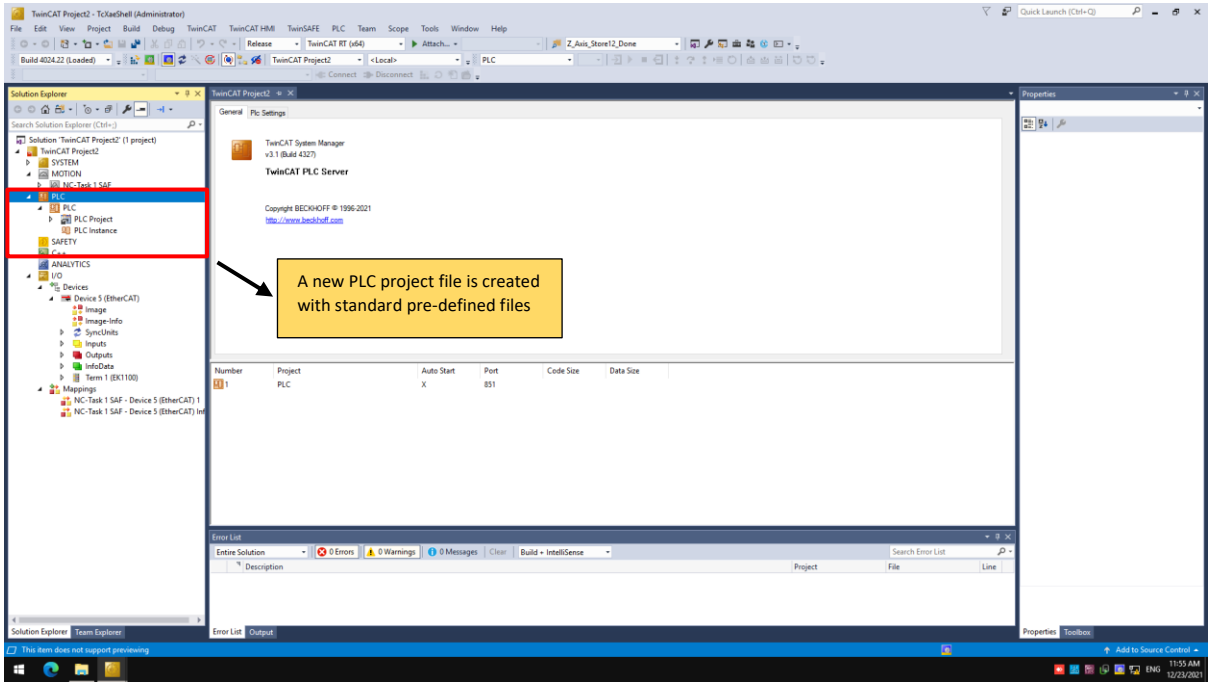


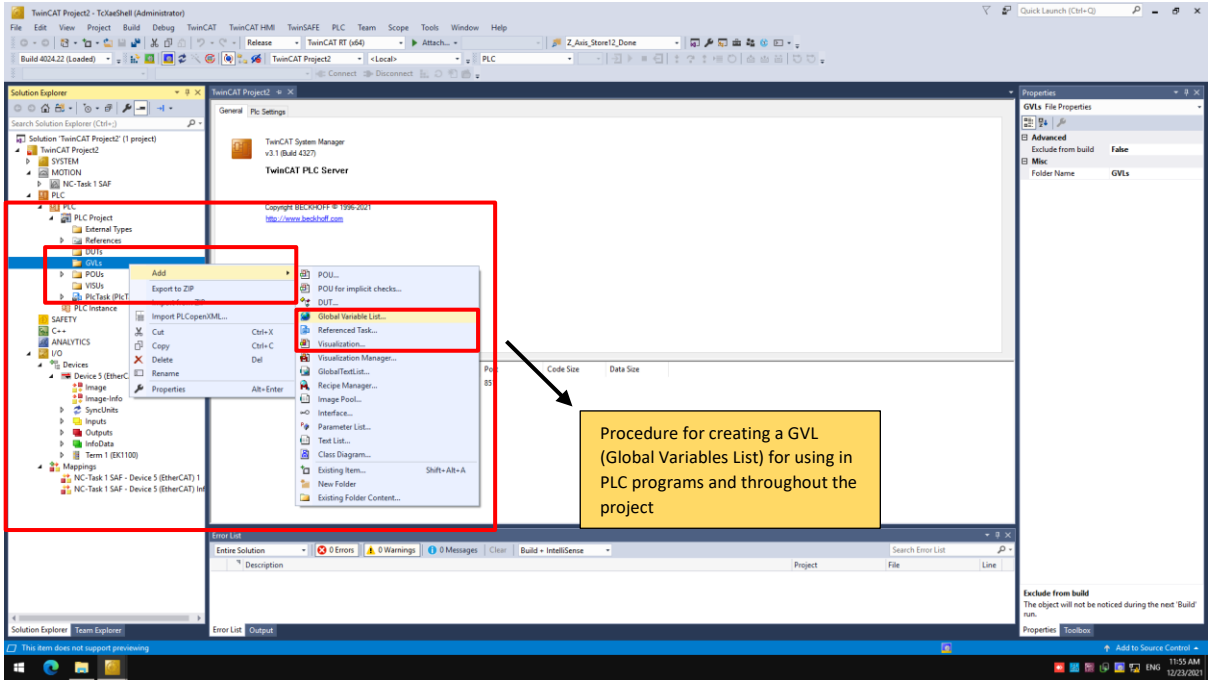


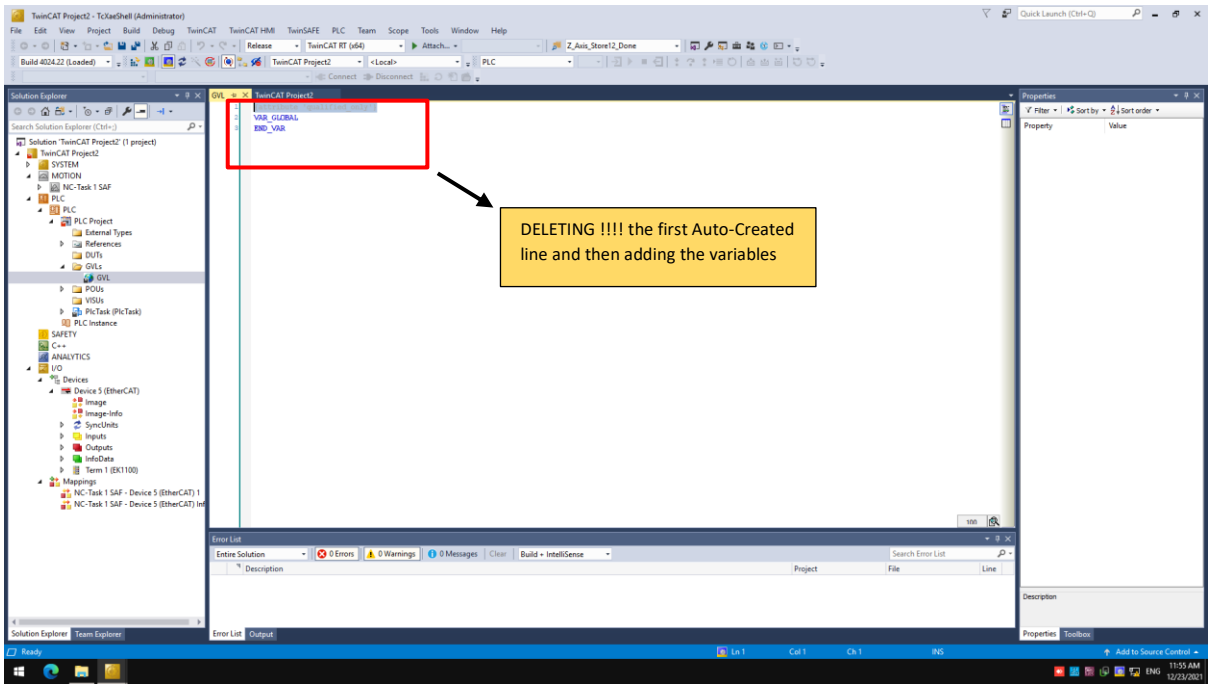
4.1.5 Creating PLC project and programs

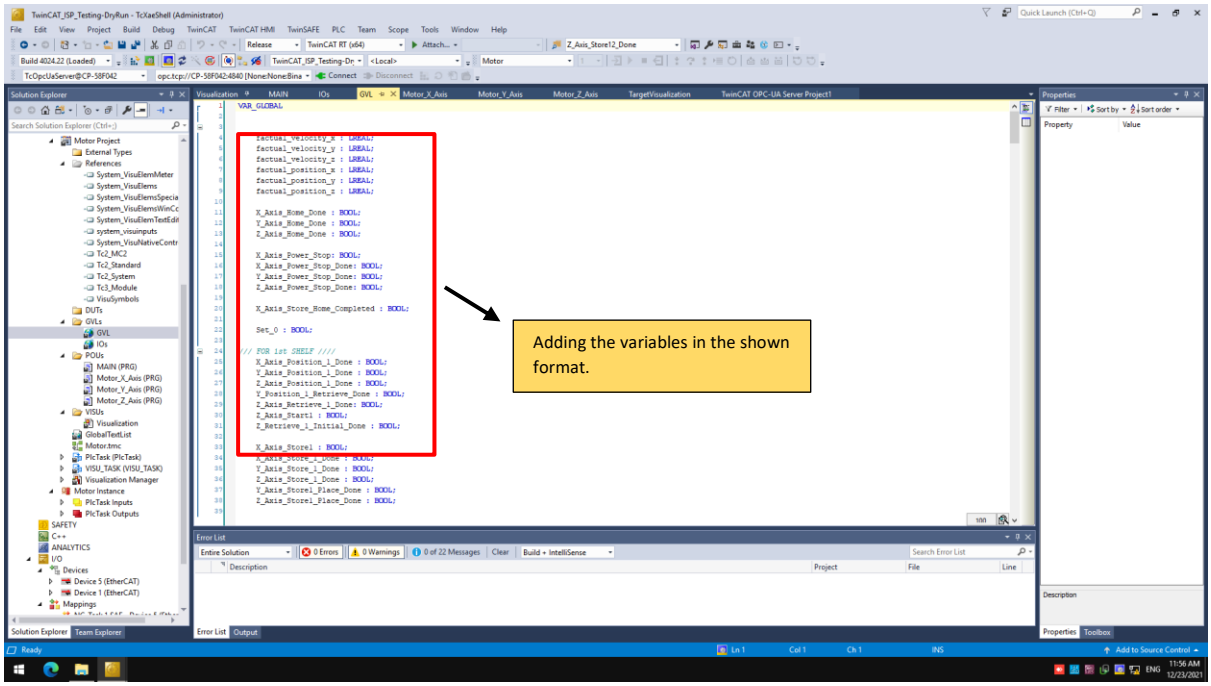




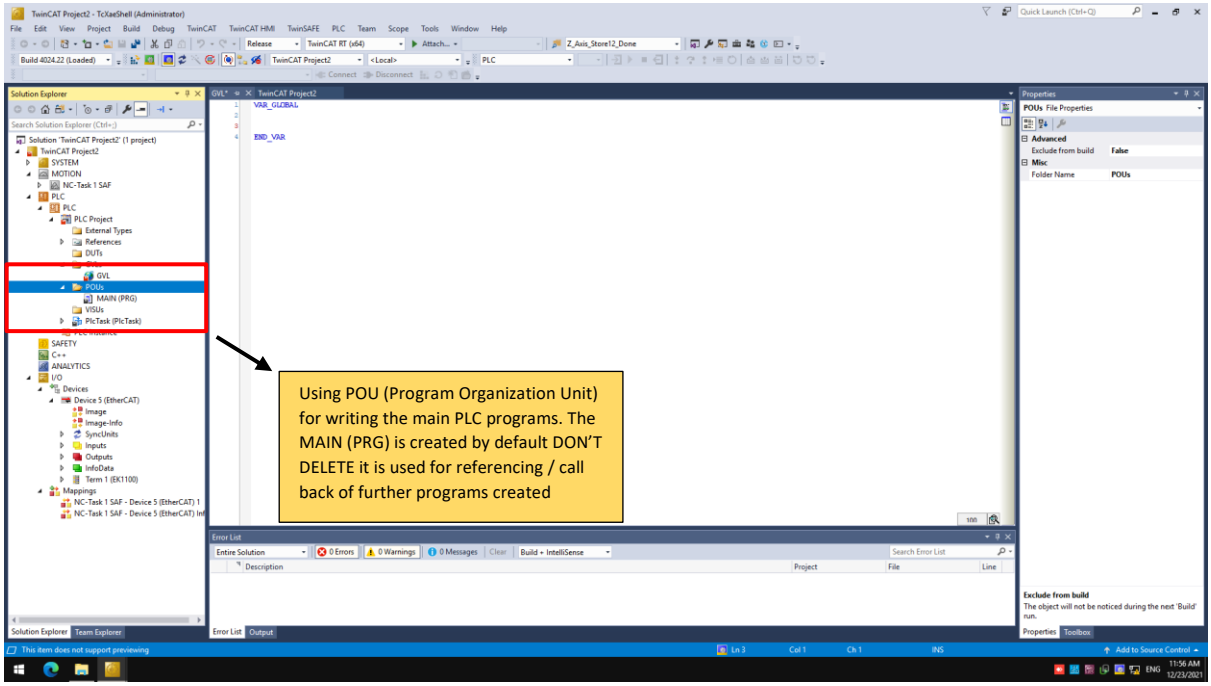

A new PLC project file is created with standard pre-defined files

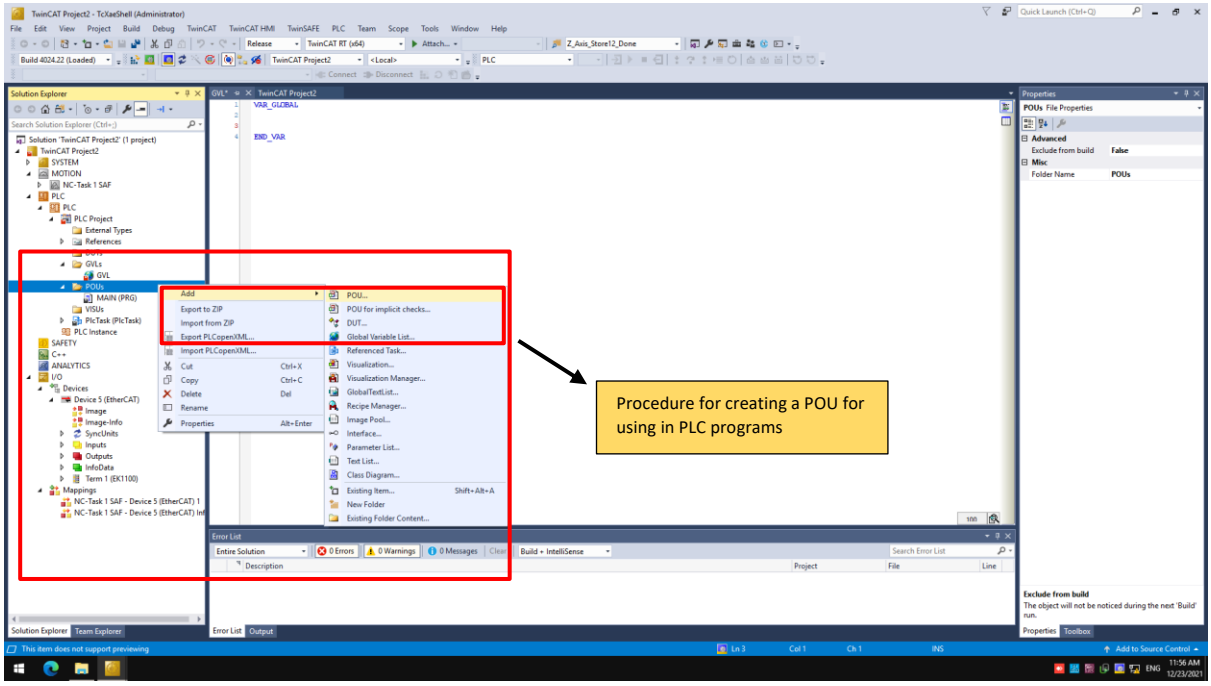

Procedure for creating a GVL (Global Variables List) for using in PLC programs and throughout the project

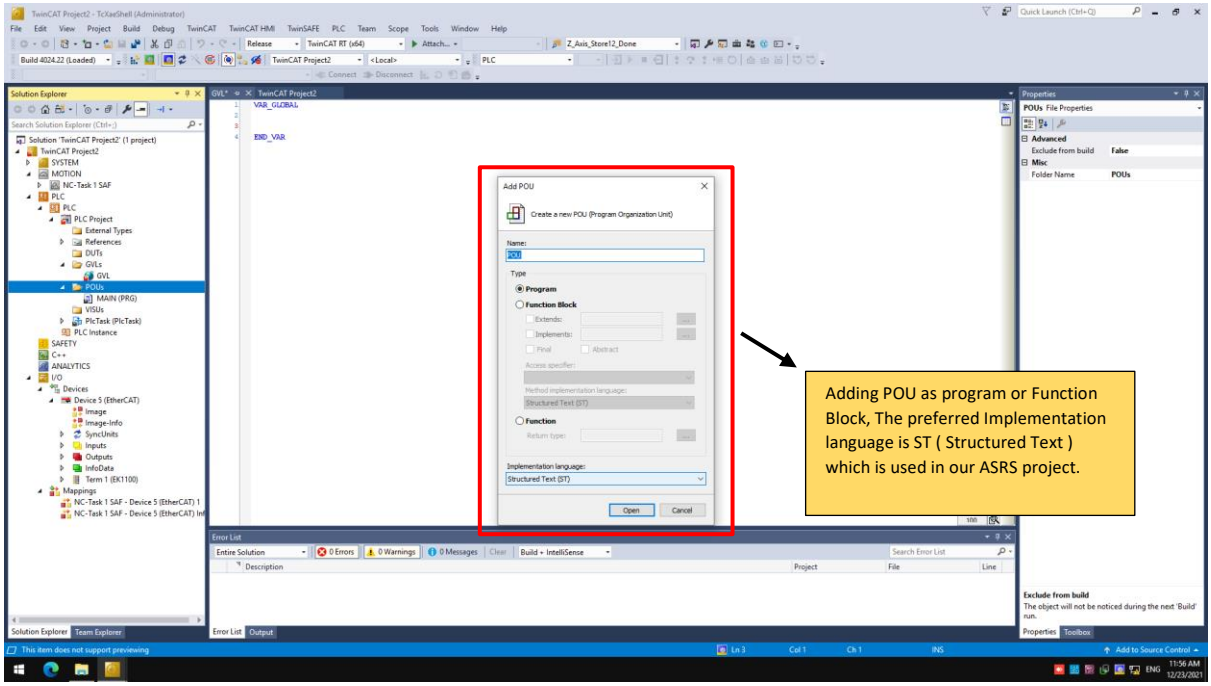

VAR_GLOBAL
END_VAR
DELETING !!!! the first Auto-Created line and then adding the variables

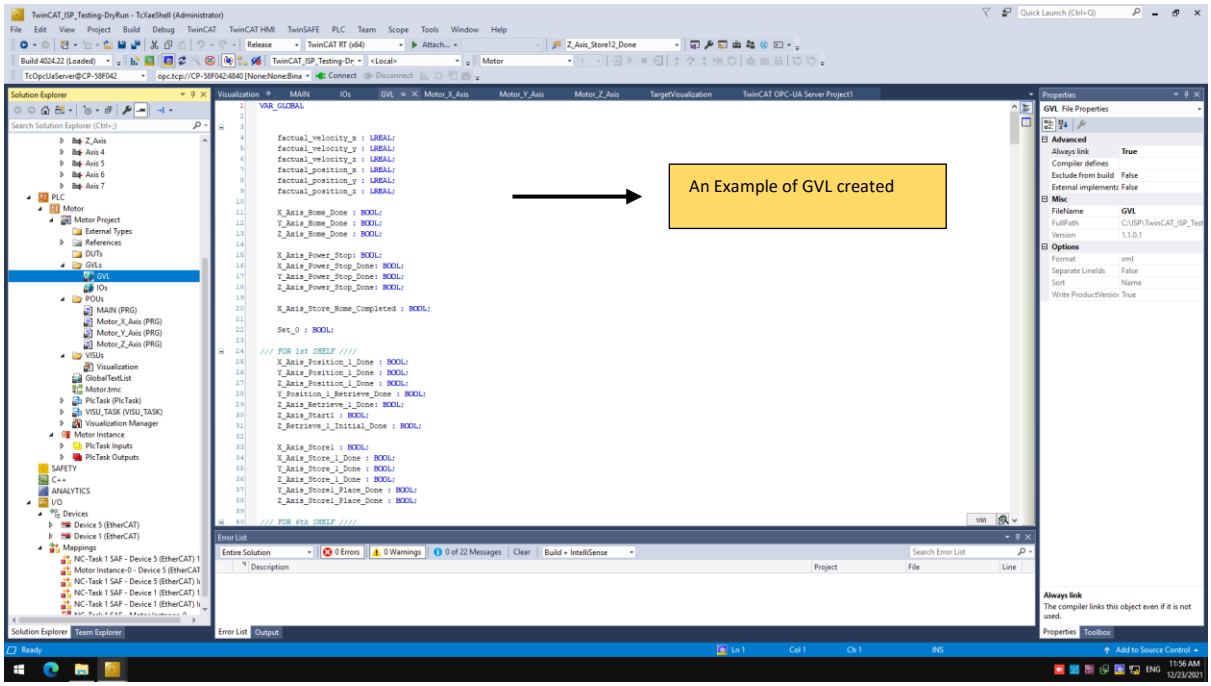

VAR_GLOBAL
Adding the variables in the shown format.

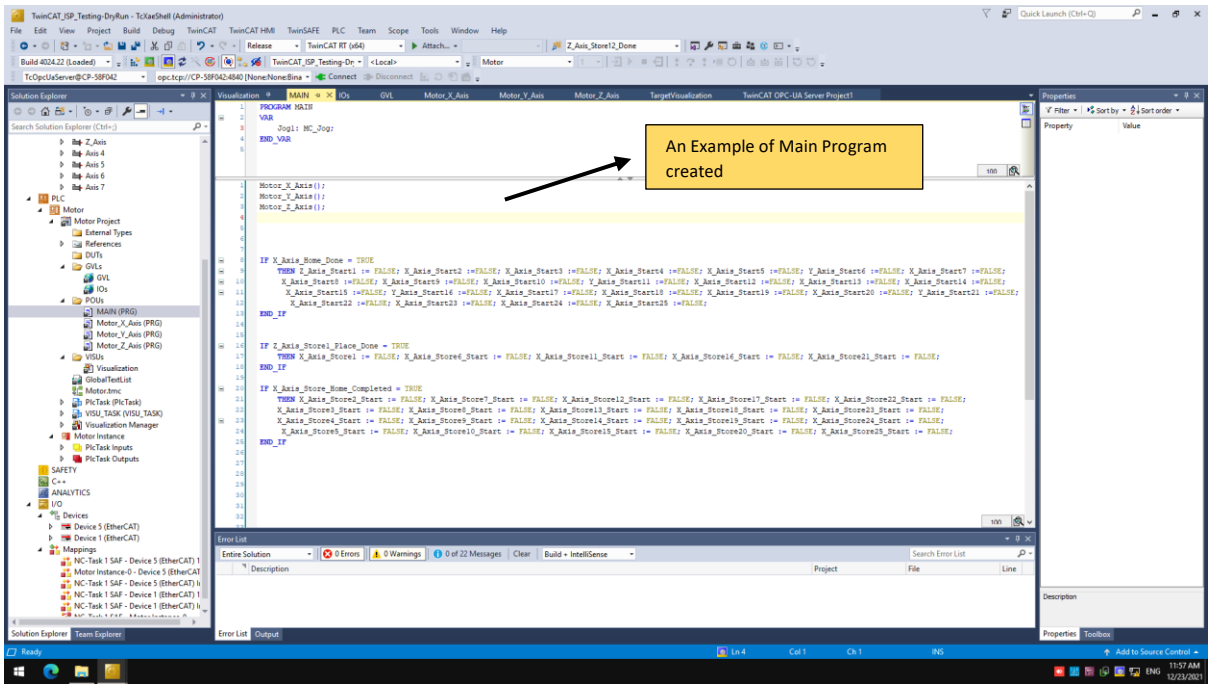
4.1.5.1 Creating program files and implementation procedures

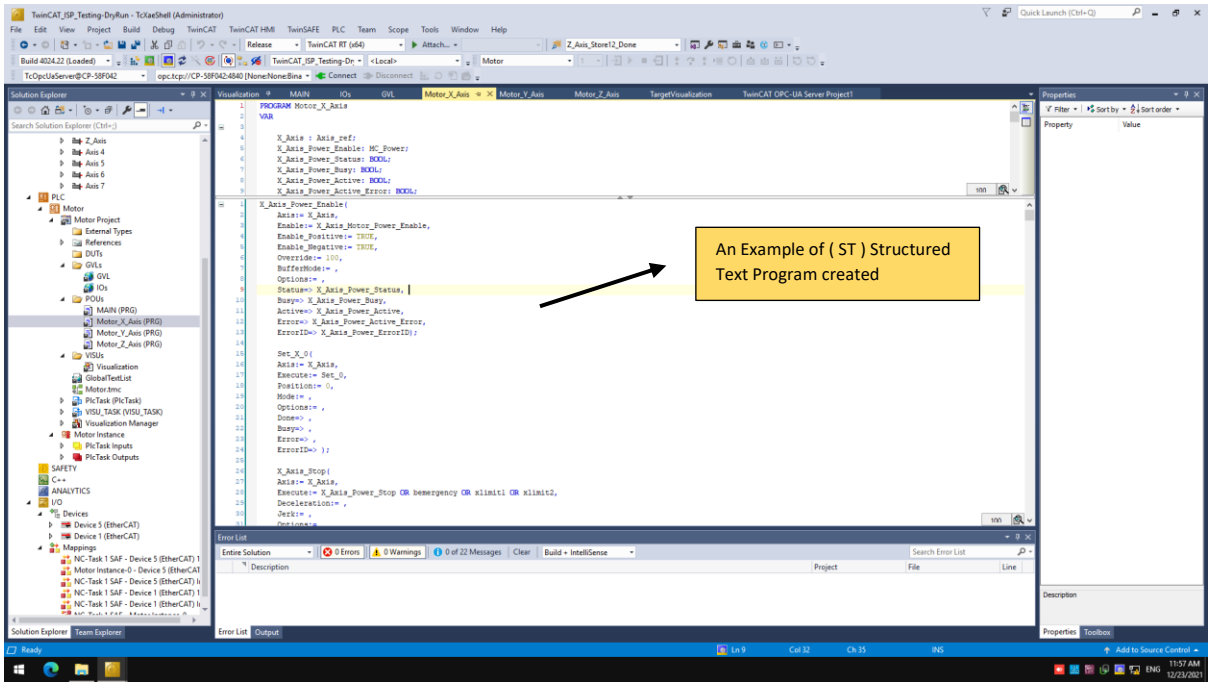




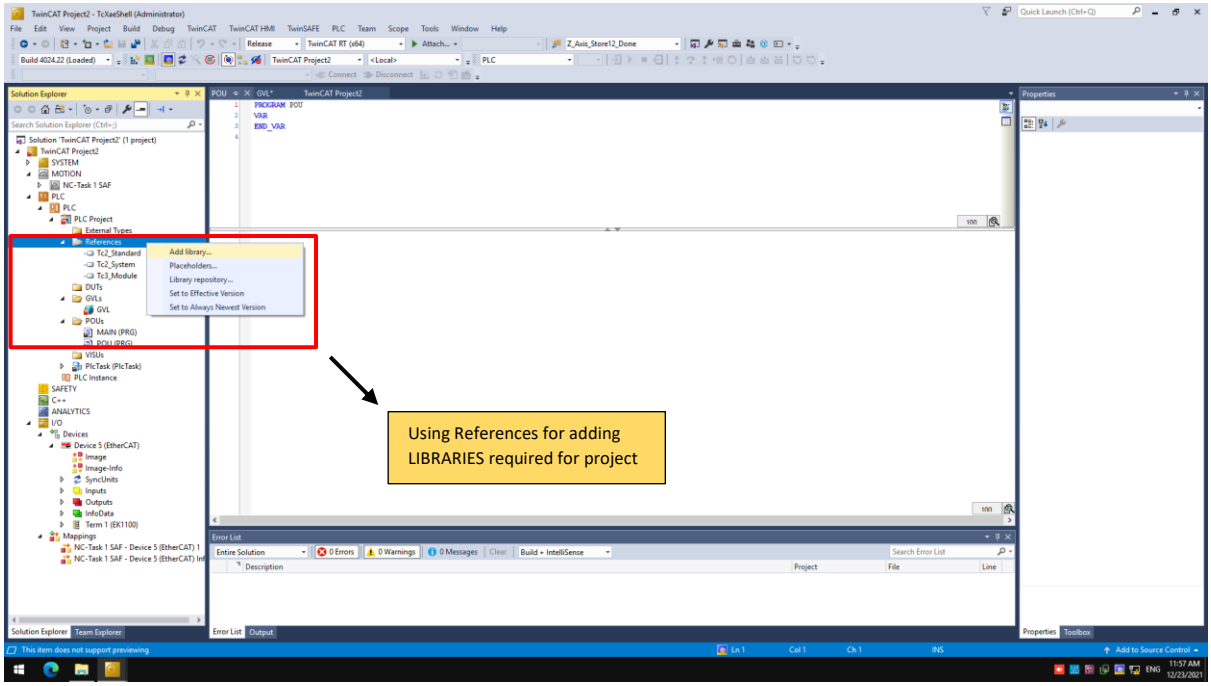

Add POU
Create a new POU (Program Organization Unit)
Name:
POU
Type
Program
Function Block
Function
Implementation language:
Structured Text (ST)
Open
Cancel
Adding POU as program or Function Block, The preferred Implementation language is ST (Structured Text) which is used in our ASRS project.

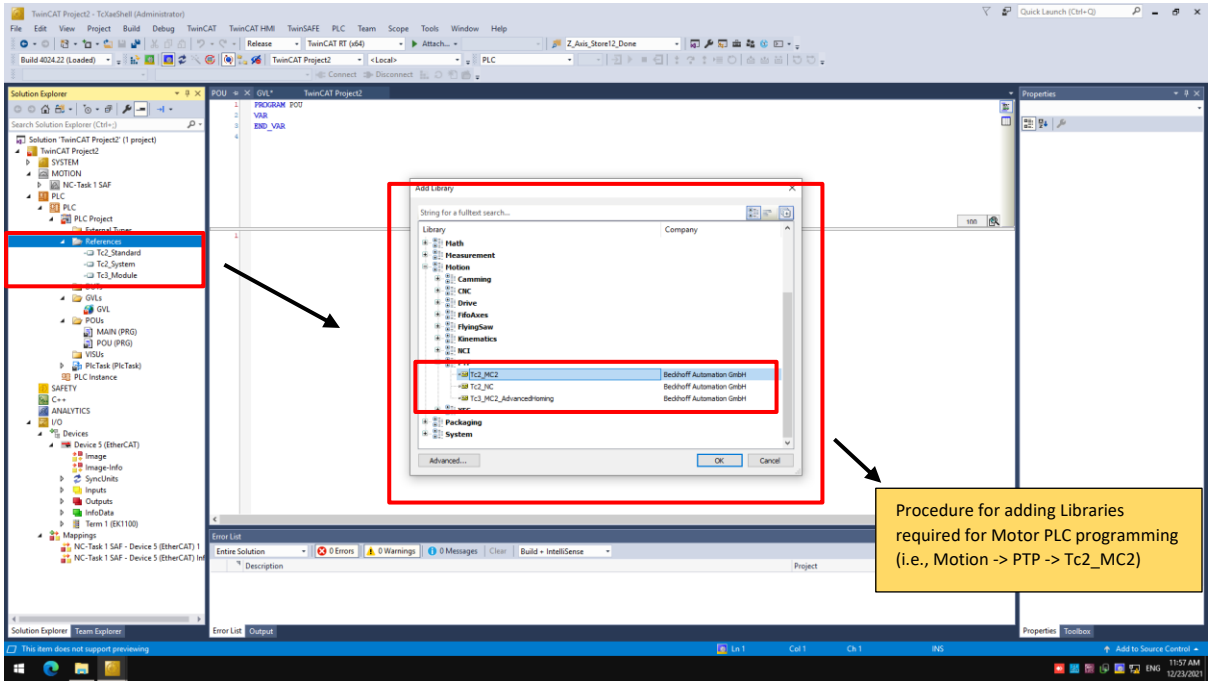

VAR_GLOBAL
factual_velocity_x : LREAL;
factual_velocity_y : LREAL;
factual_velocity_z : LREAL;
factual_position_x : LREAL;
factual_position_y : LREAL;
factual_position_z : LREAL;
X_Axis_Home_Done : BOOL;
Y_Axis_Home_Done : BOOL;
Z_Axis_Home_Done : BOOL;
X_Axis_Power_Stop: BOOL;
X_Axis_Power_Stop_Done: BOOL;
Y_Axis_Power_Stop_Done: BOOL;
Z_Axis_Power_Stop_Done: BOOL;
X_Axis_Store_Home_Completed : BOOL;
Set_0 : BOOL;
/// FOR 1st SHELF ////
X_Axis_Position_1_Done : BOOL;
Y_Axis_Position_1_Done : BOOL;
Z_Axis_Position_1_Done : BOOL;
Y_Position_1_Retrieve_Done : BOOL;
Z_Axis_Retrieve_1_Done: BOOL;
Z_Axis_Start1 : BOOL;
Z_Retrieve_1_Initial_Done : BOOL;
X_Axis_Store1 : BOOL;
X_Axis_Store_1_Done : BOOL;
Y_Axis_Store_1_Done : BOOL;
Z_Axis_Store_1_Done : BOOL;
Y_Axis_Store1_Place_Done : BOOL;
Z_Axis_Store1_Place_Done : BOOL;
/// FOR 4th SHELF ////
An Example of GVL created

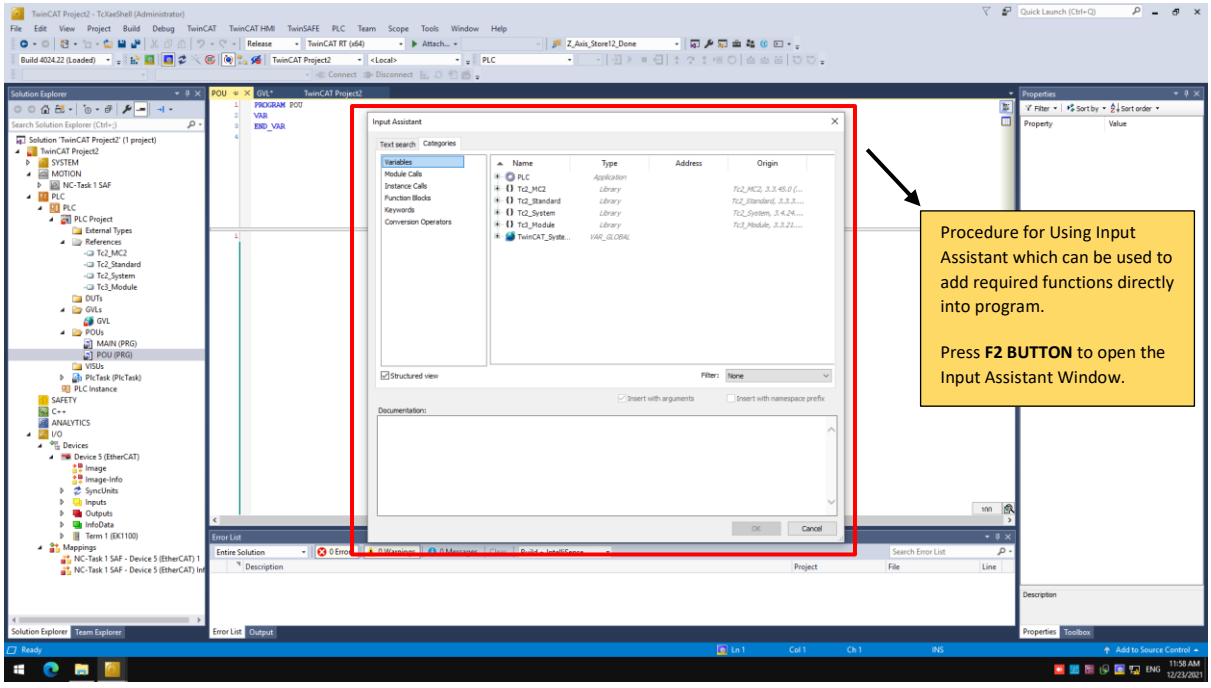

An Example of Main Program created

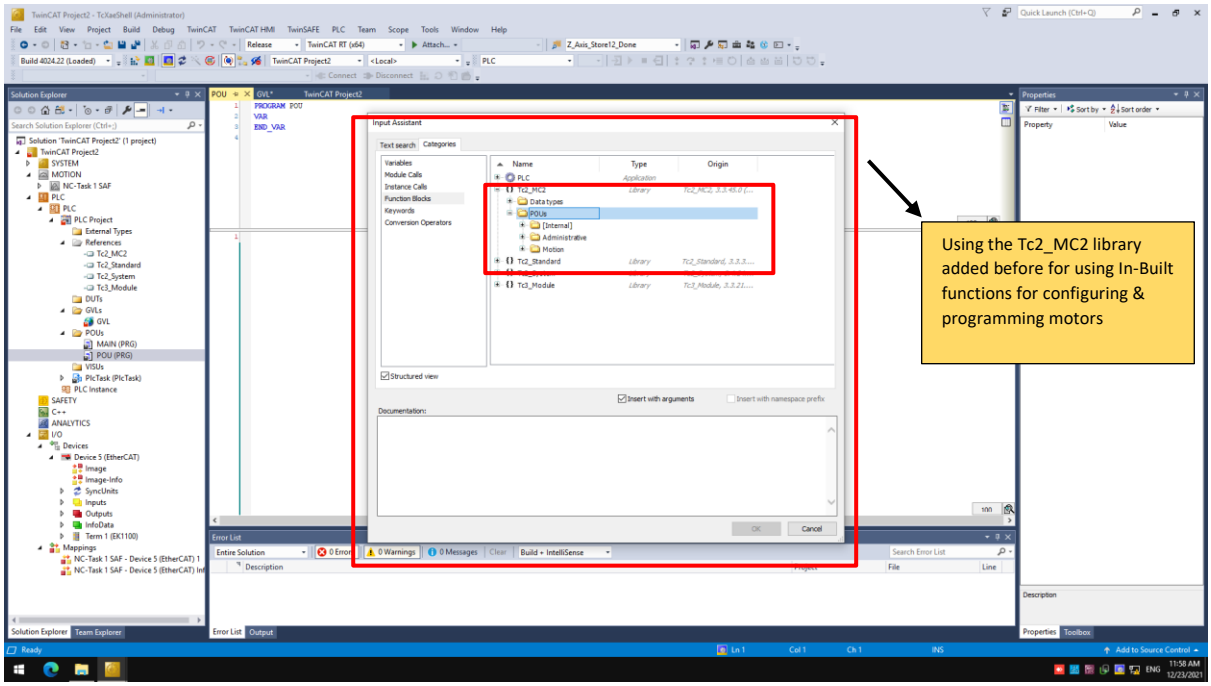

An Example of (ST) Structured Text Program created

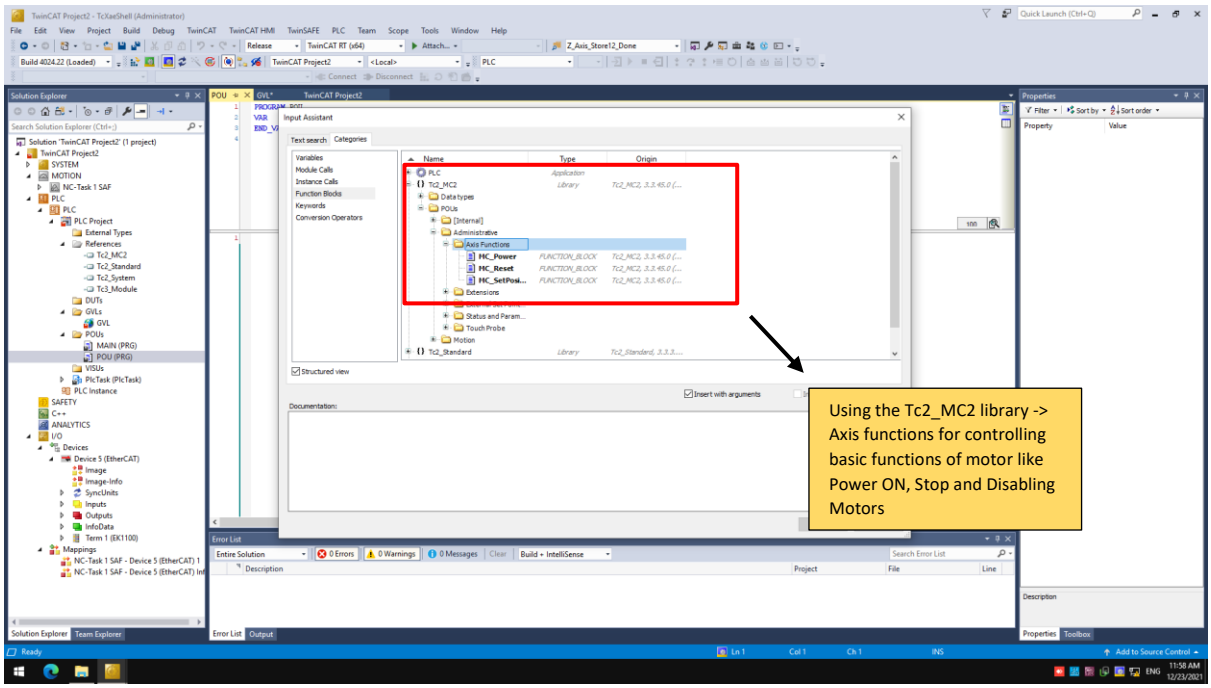
4.1.5.2 Adding Libraries and using In-built functions

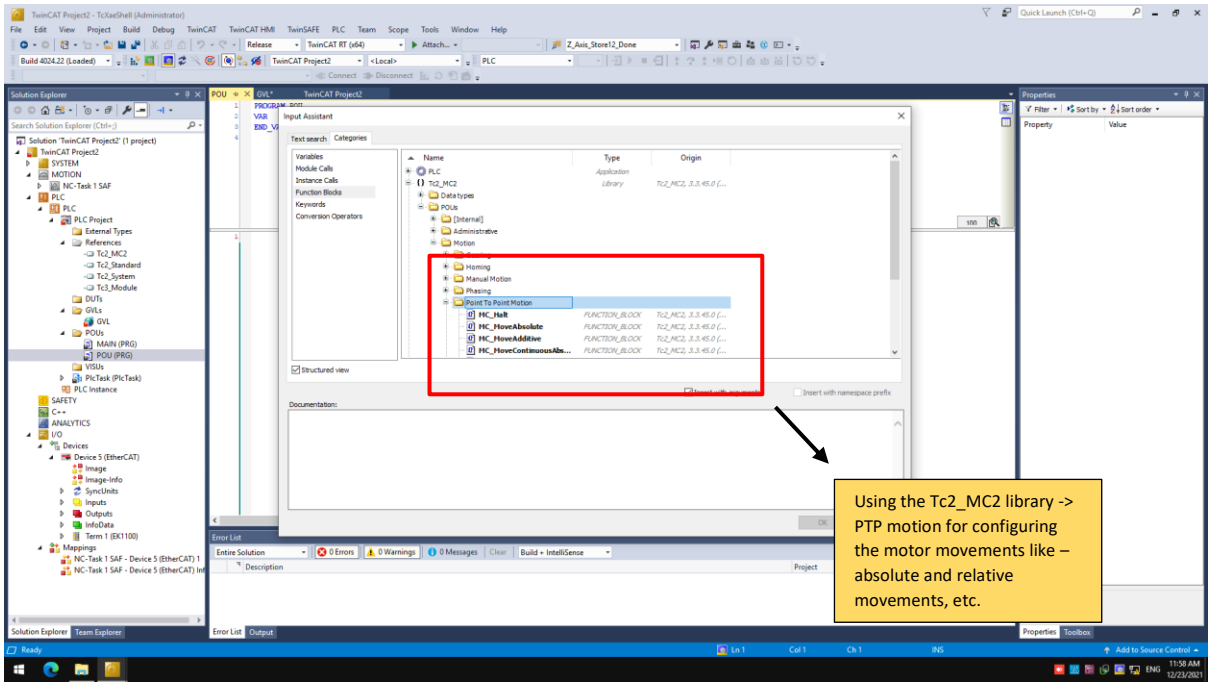


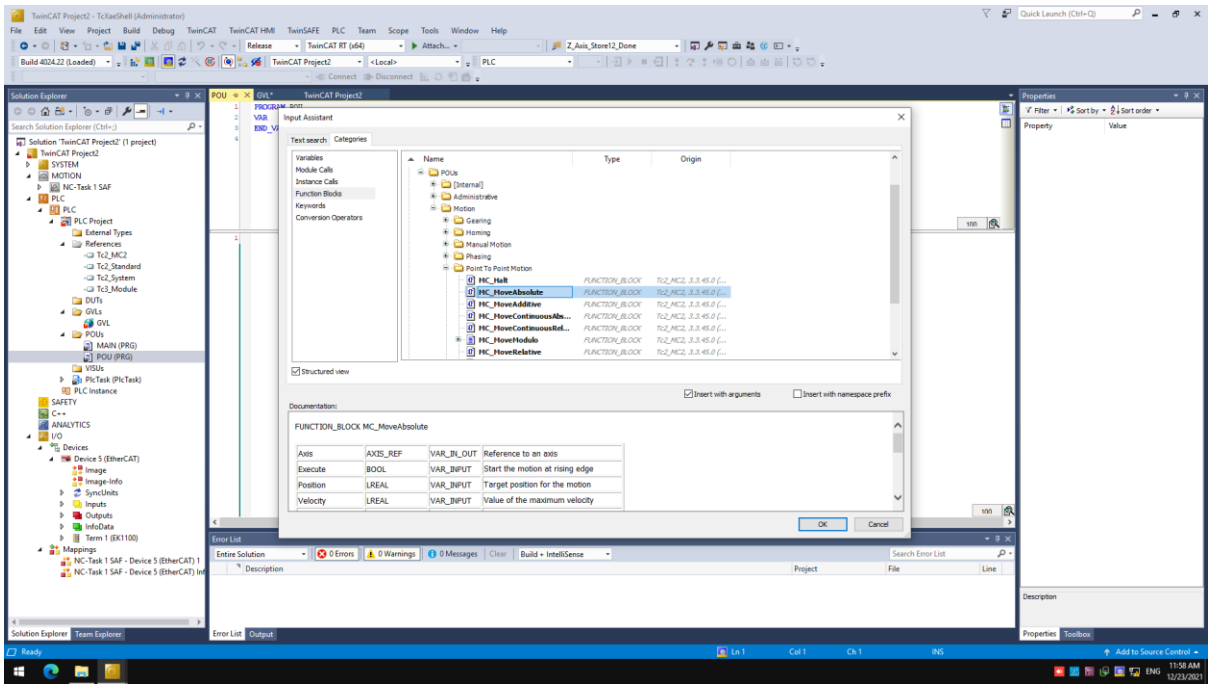


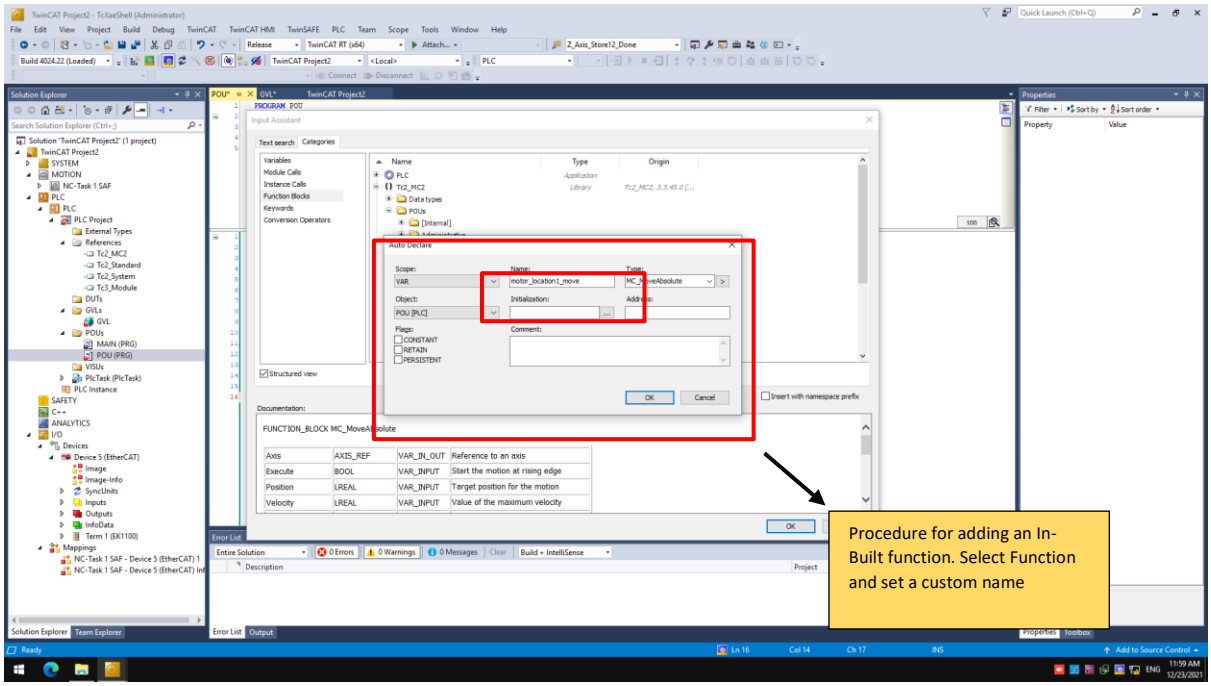

Procedure for Using Input Assistant which can be used to add required functions directly into program.
Press F2 BUTTON to open the Input Assistant Window.

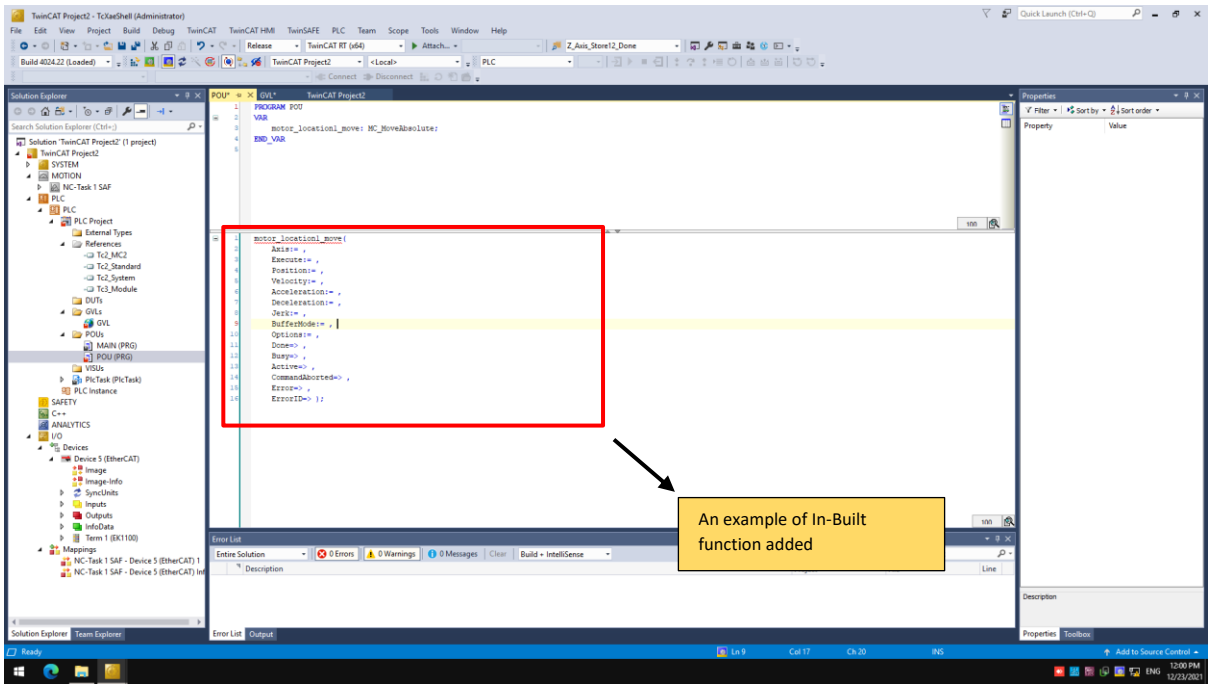

Using the Tc2_MC2 library added before for using In-Built functions for configuring & programming motors

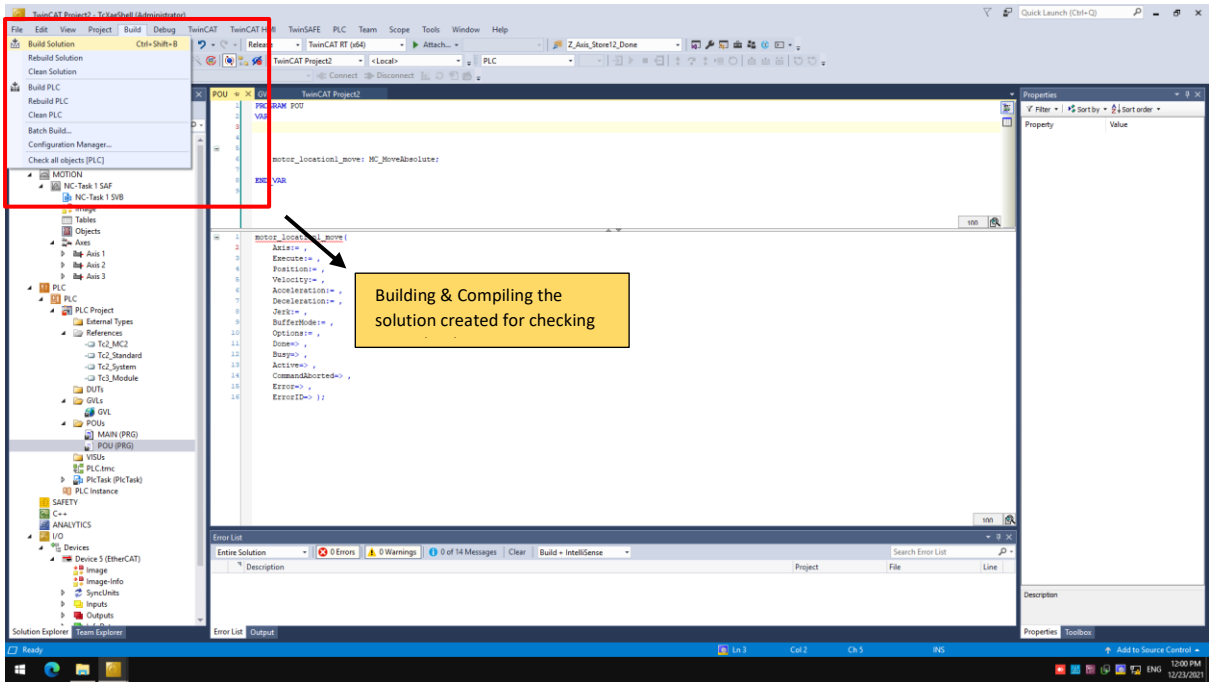

Using the Tc2_MC2 library -> Axis functions for controlling basic functions of motor like Power ON, Stop and Disabling Motors

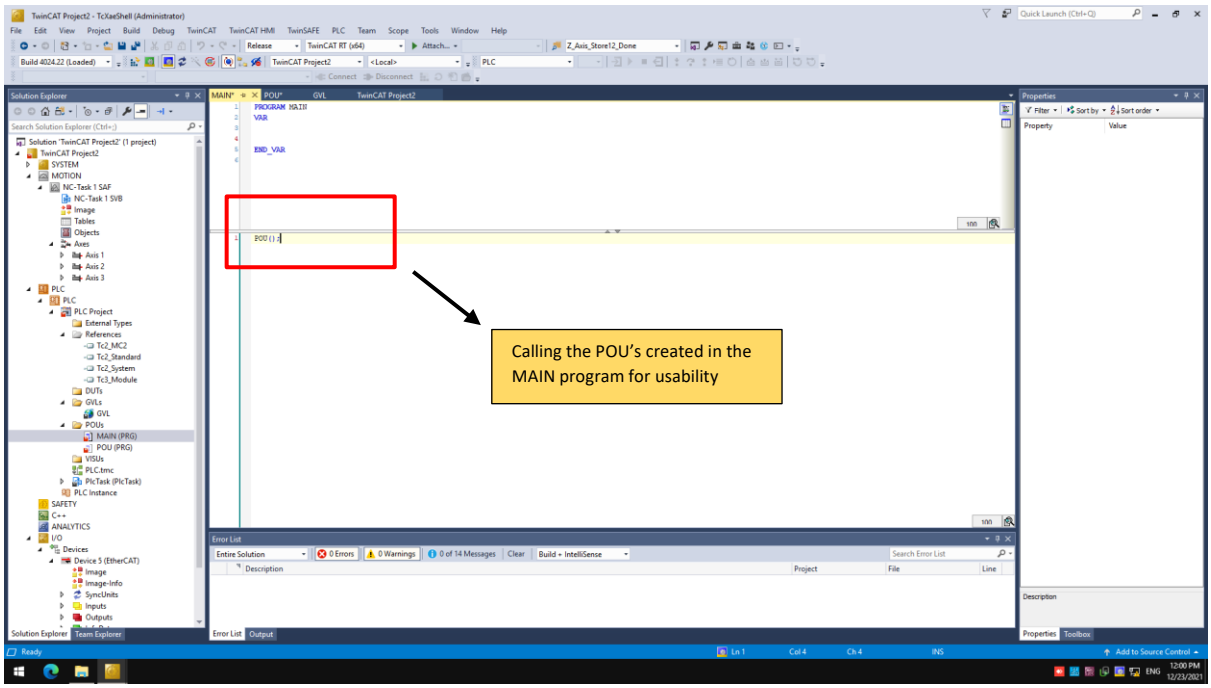

Using the Tc2_MC2 library -> PTP motion for configuring the motor movements like – absolute and relative movements, etc.

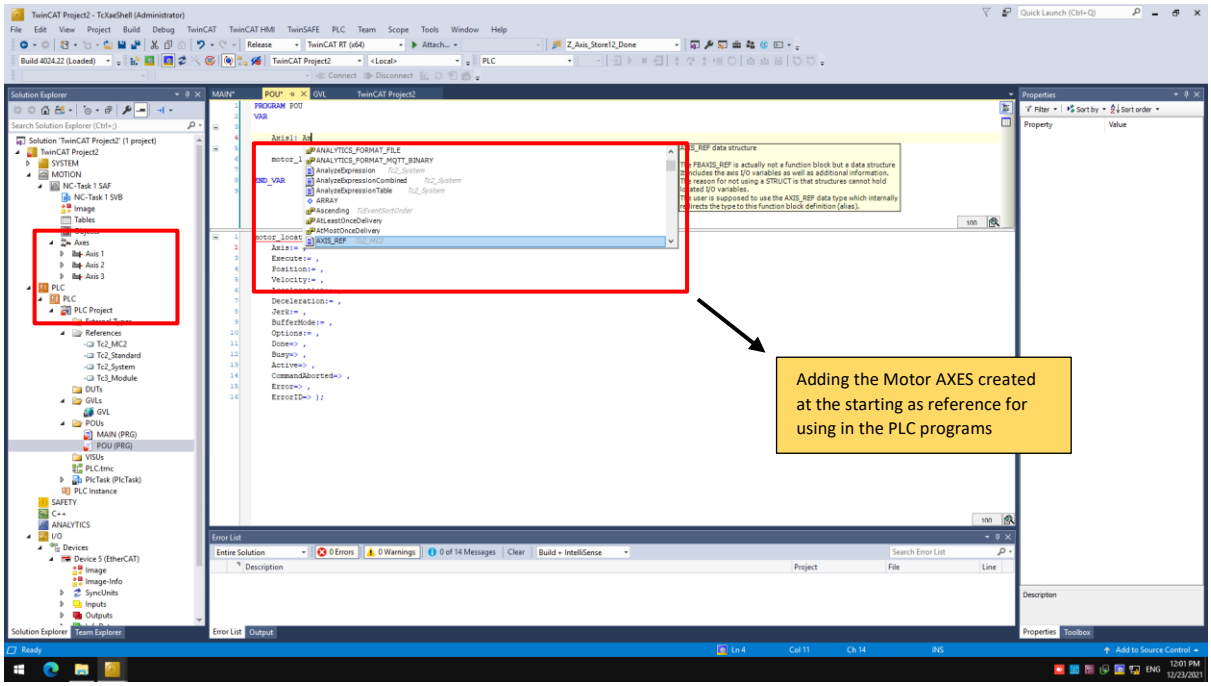


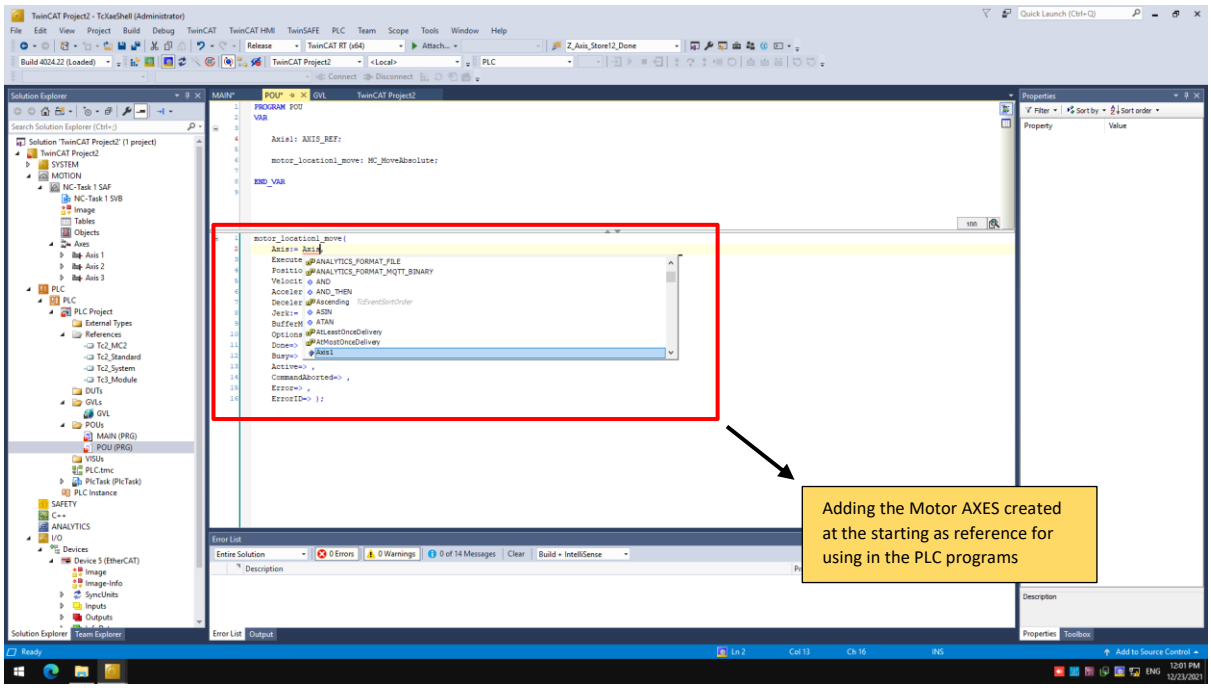

Procedure for adding an In-Built function. Select Function and set a custom name

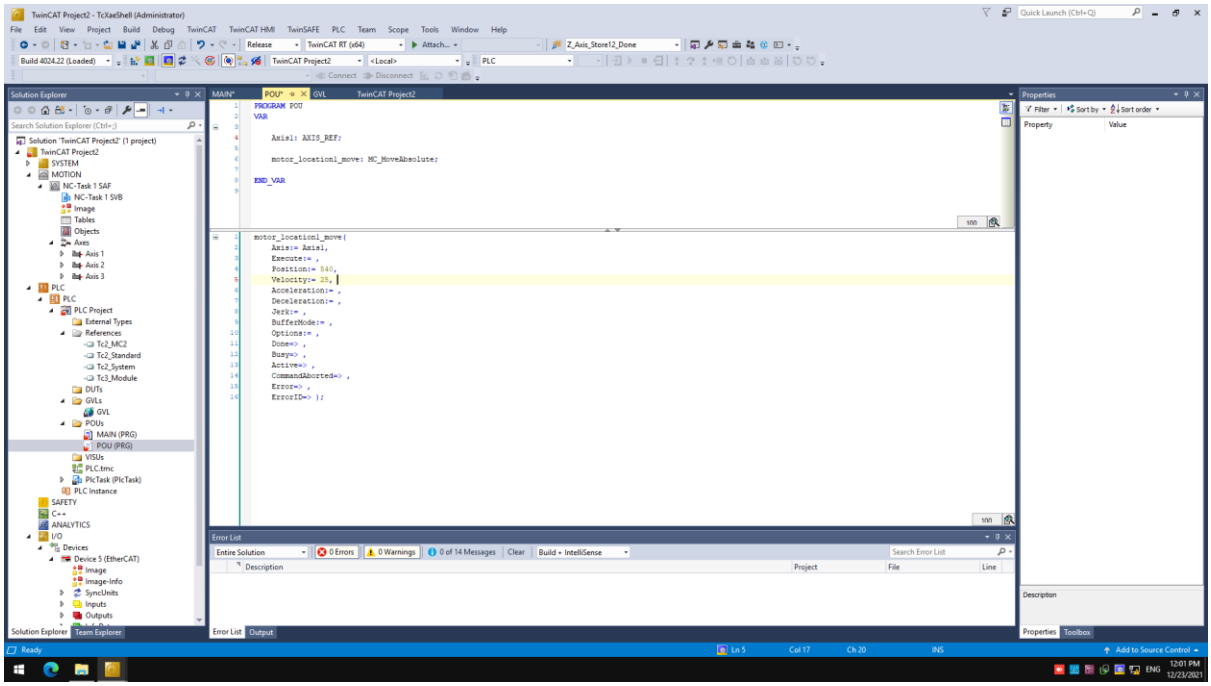

An example of In-Built function added


Building & Compiling the solution created for checking

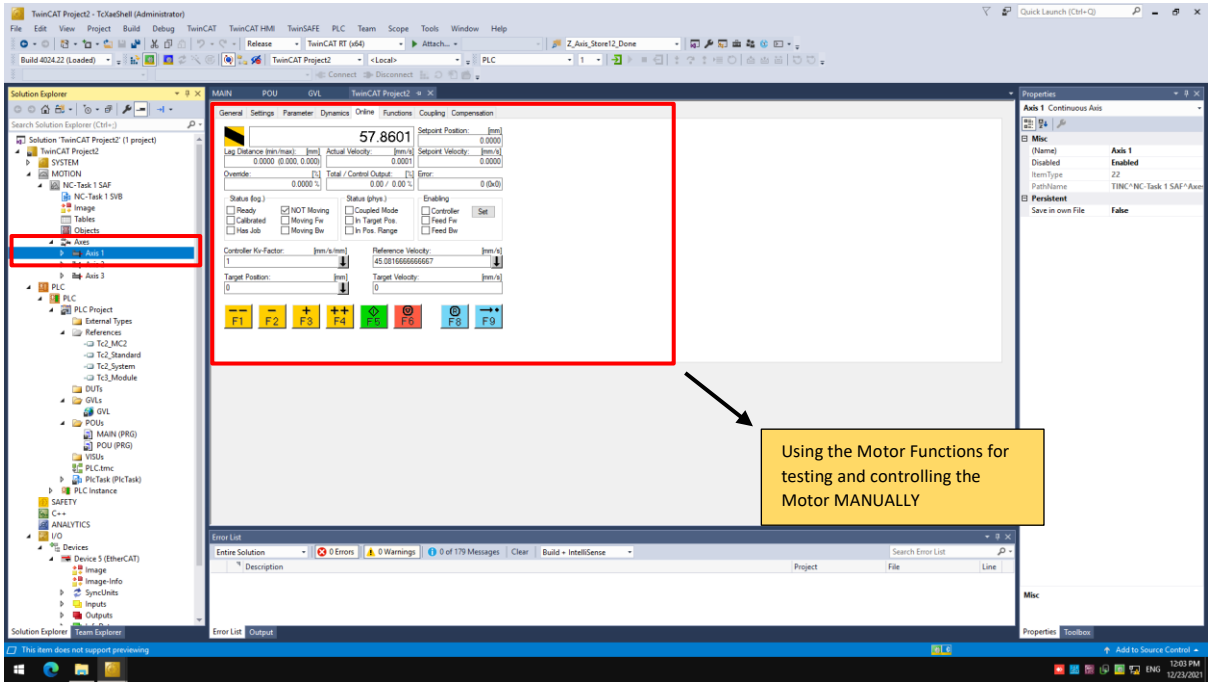

Calling the POU's created in the MAIN program for usability

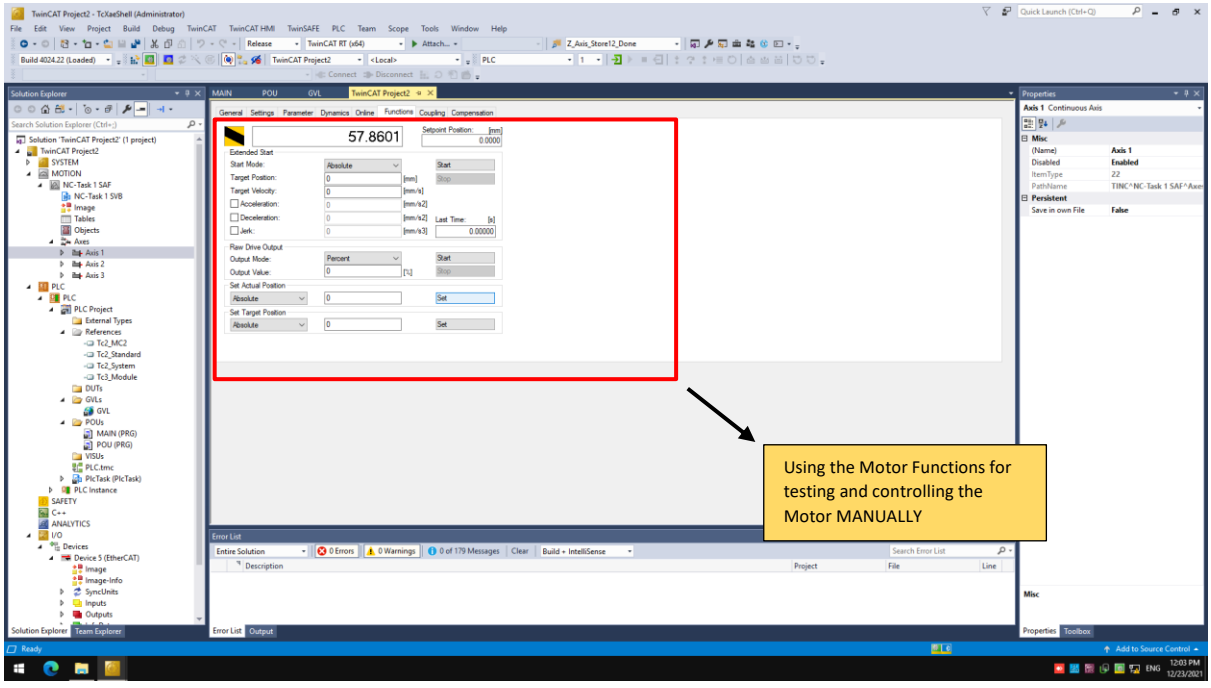

Adding the Motor AXES created at the starting as reference for using in the PLC programs

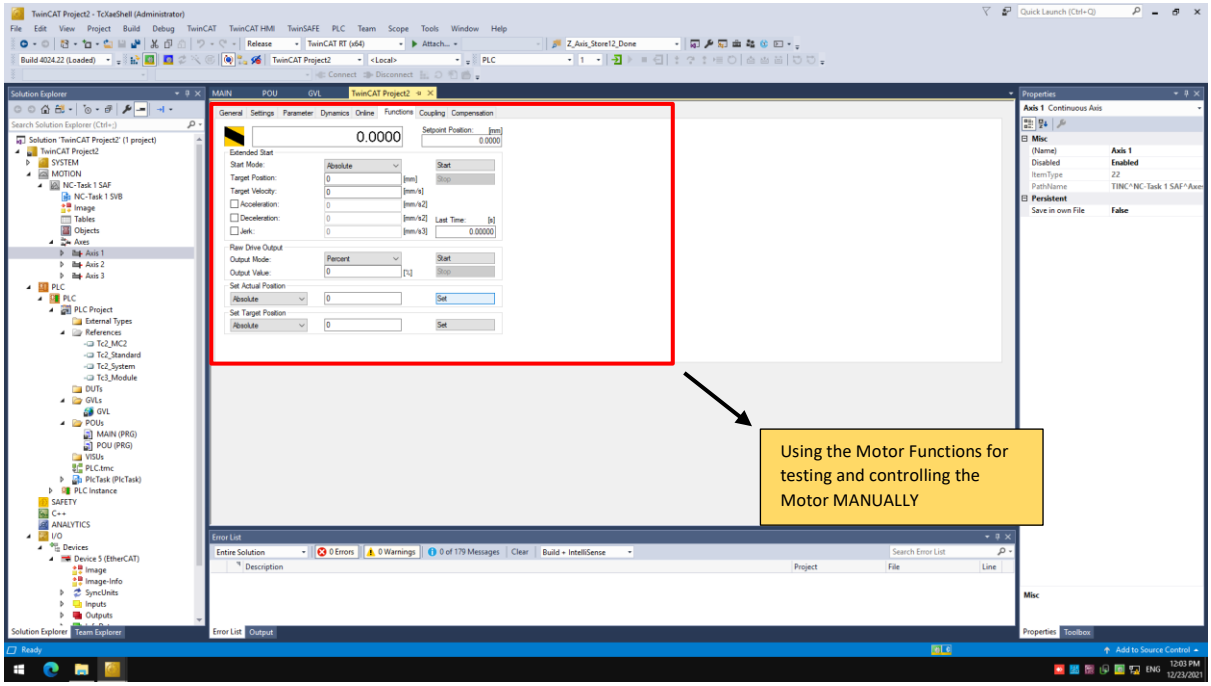

Adding the Motor AXES created at the starting as reference for using in the PLC programs

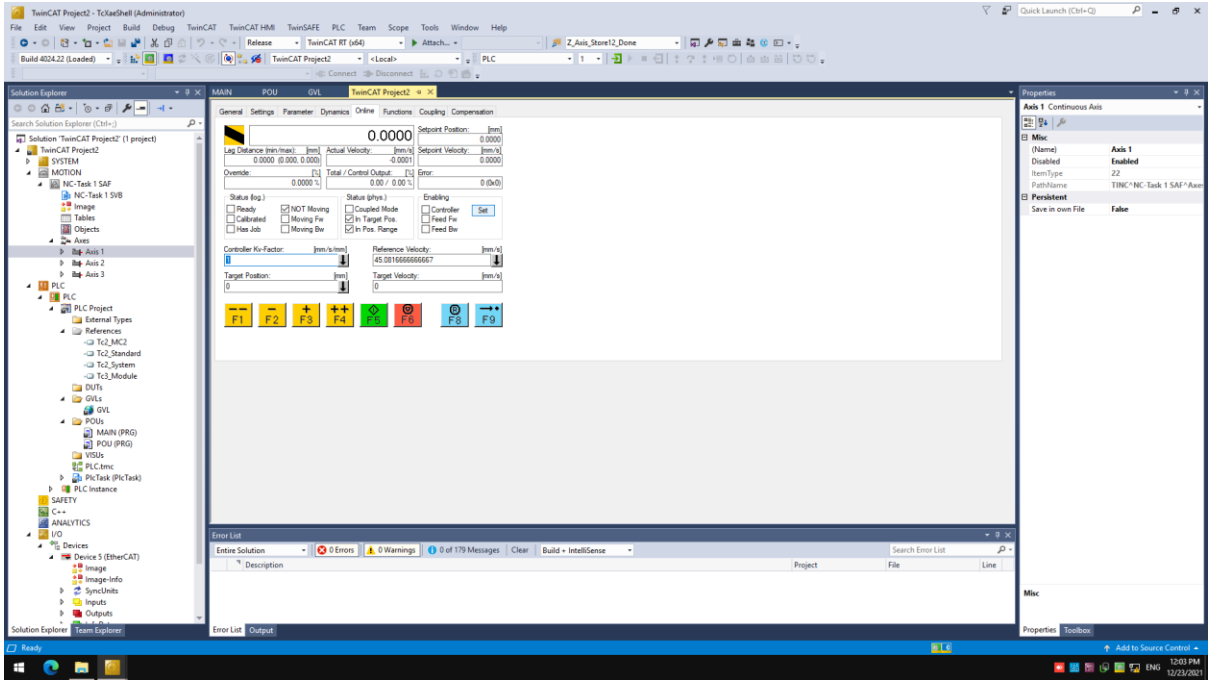


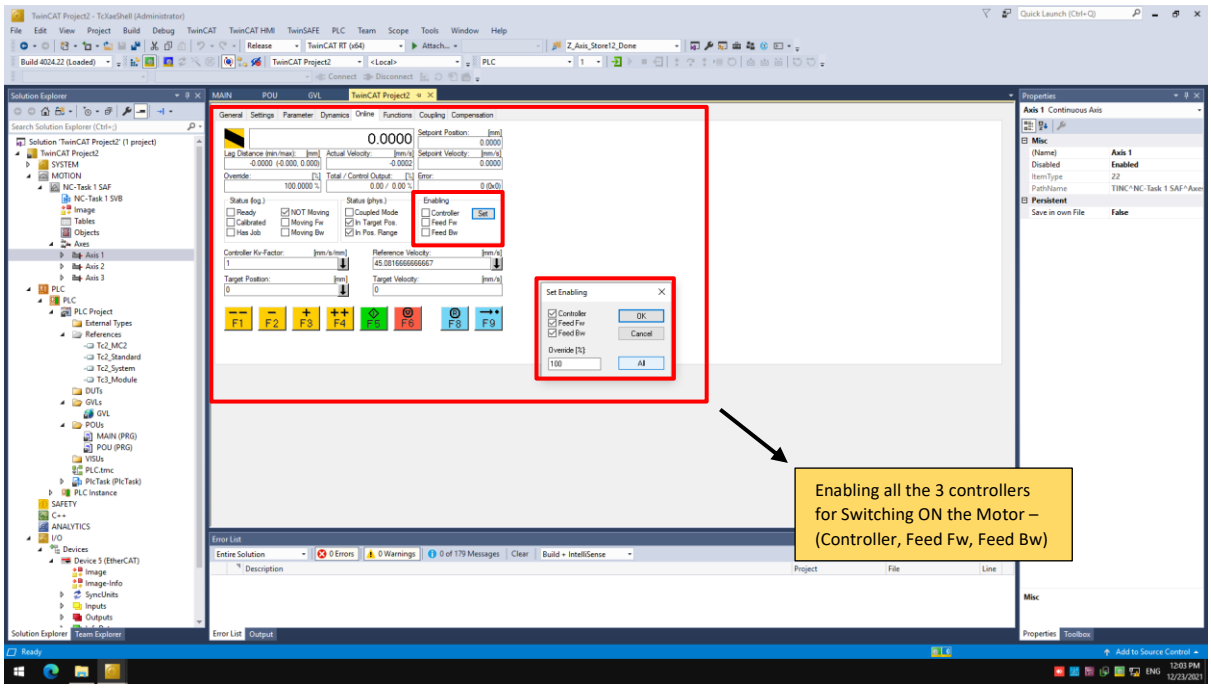
4.1.6 Controlling and testing Motors through TwinCAT interface

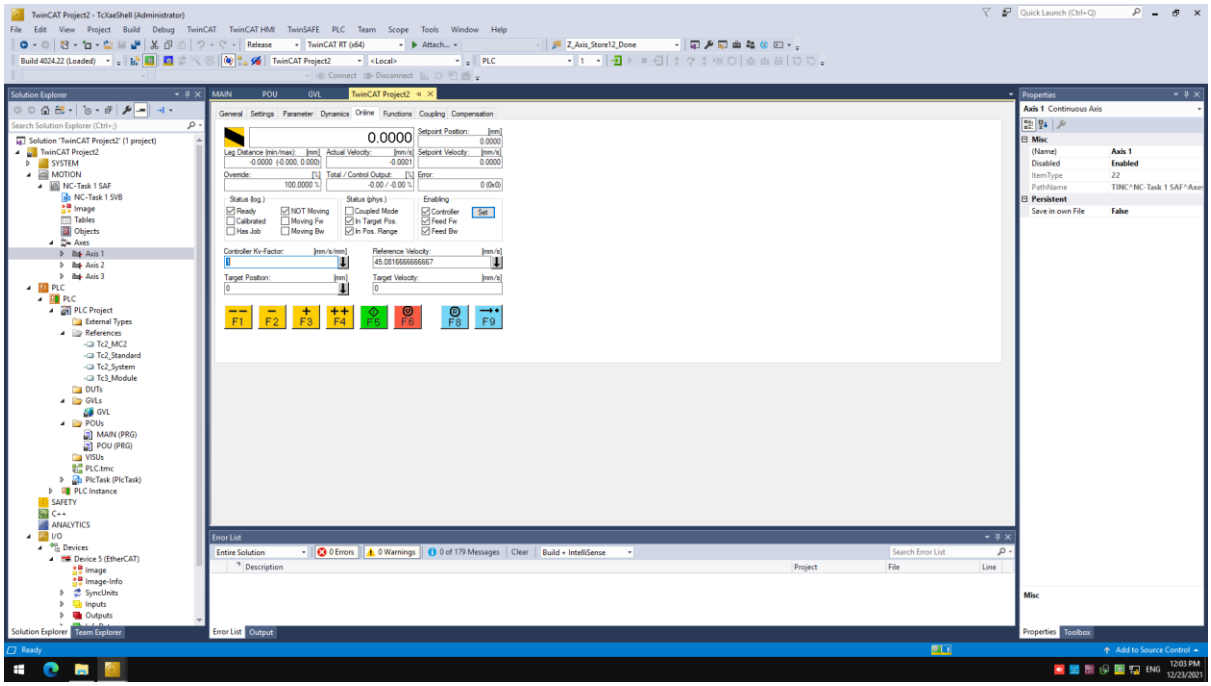


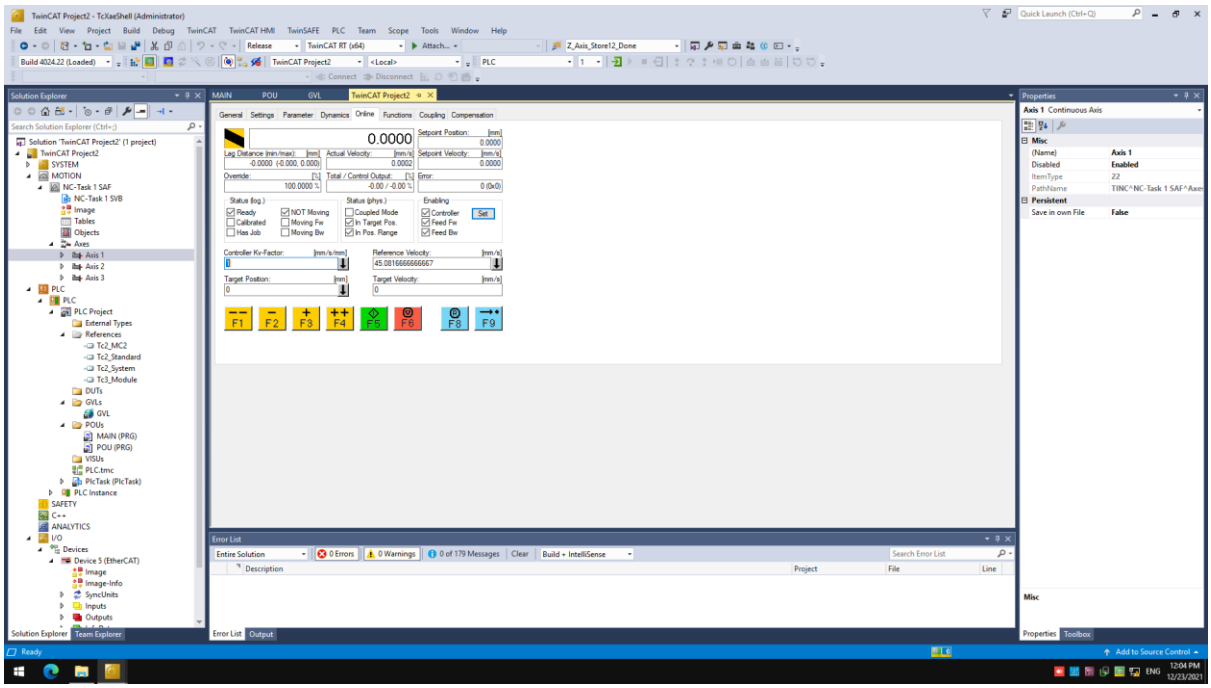


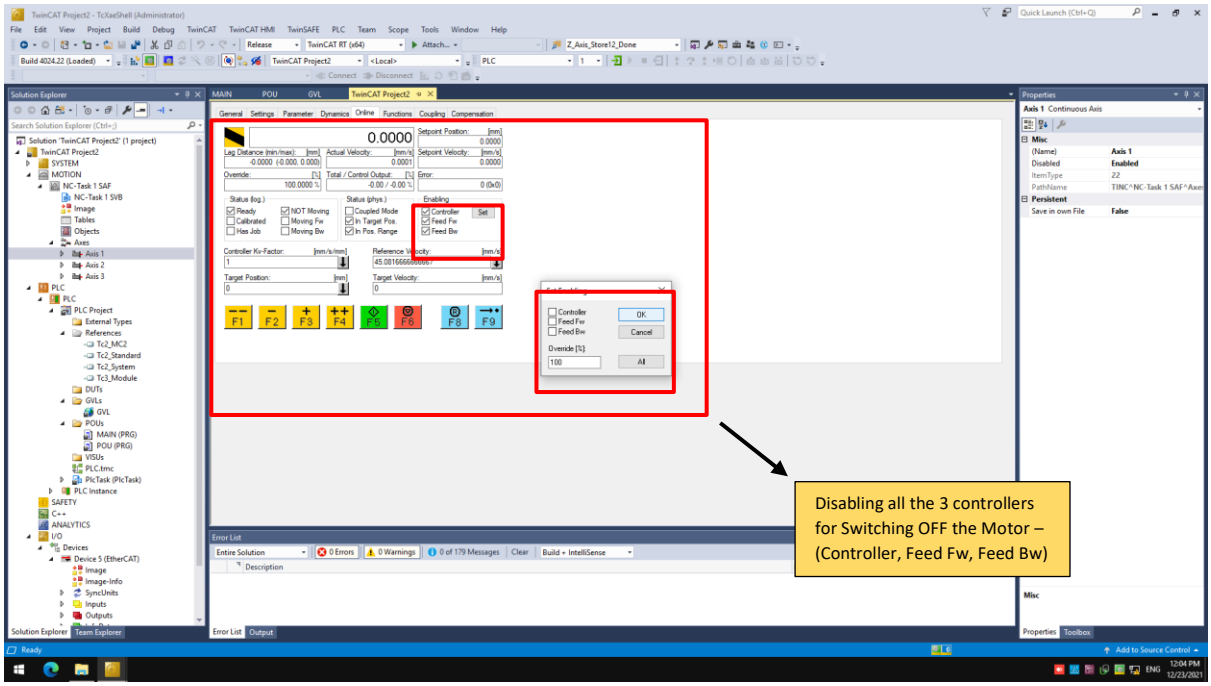

Using the Motor Functions for testing and controlling the Motor MANUALLY

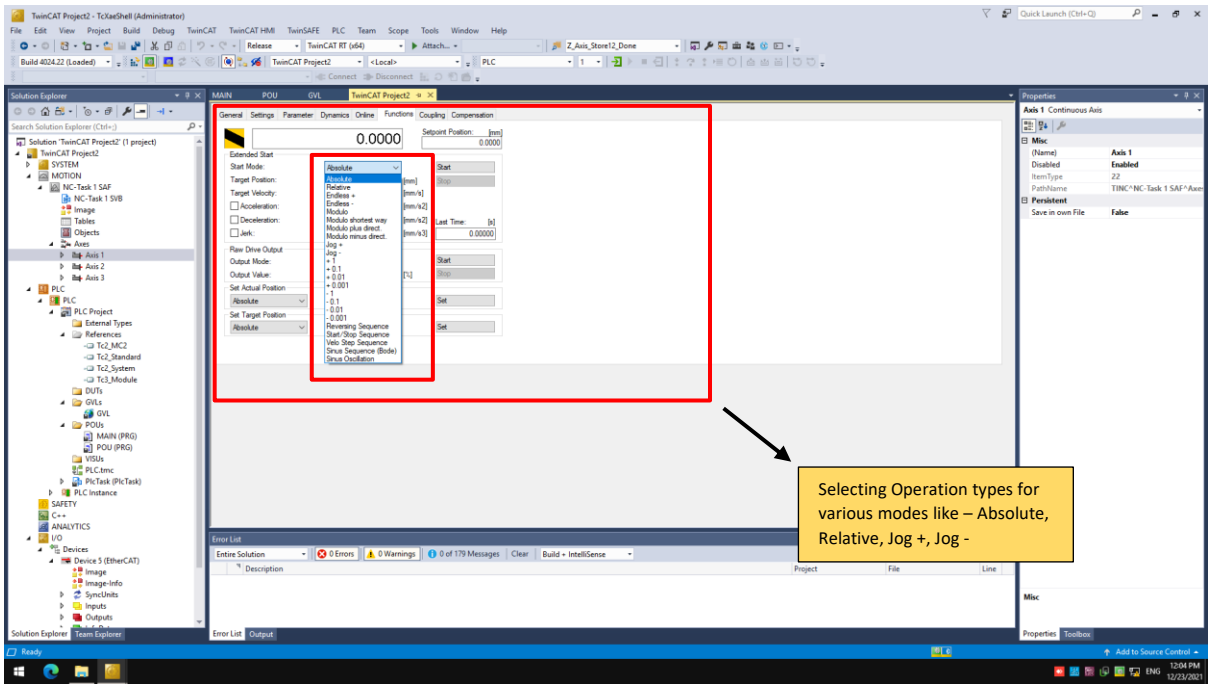


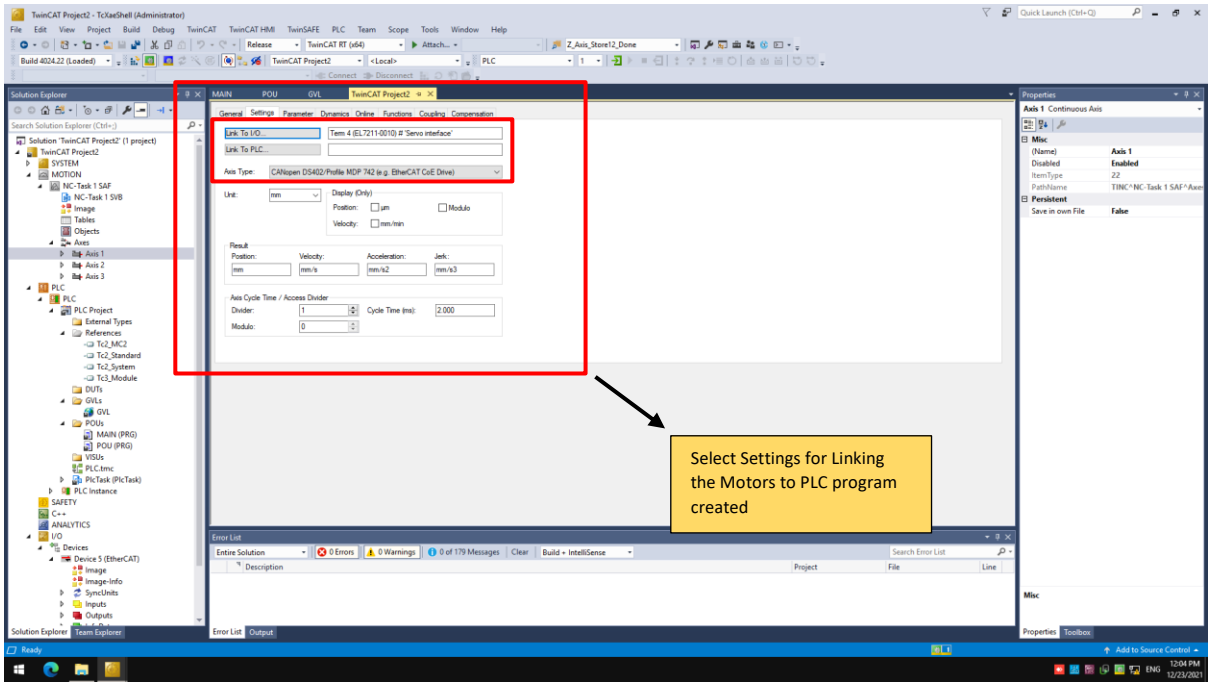

Enabling all the 3 controllers for Switching ON the Motor – (Controller, Feed Fw, Feed Bw)



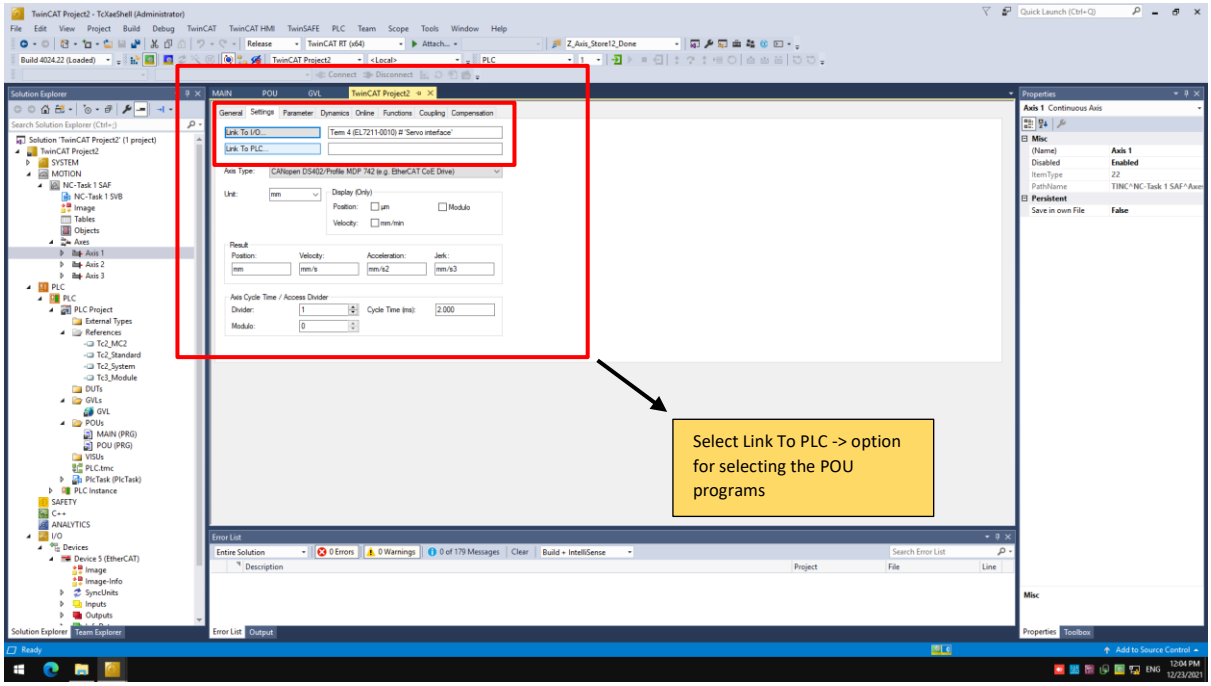


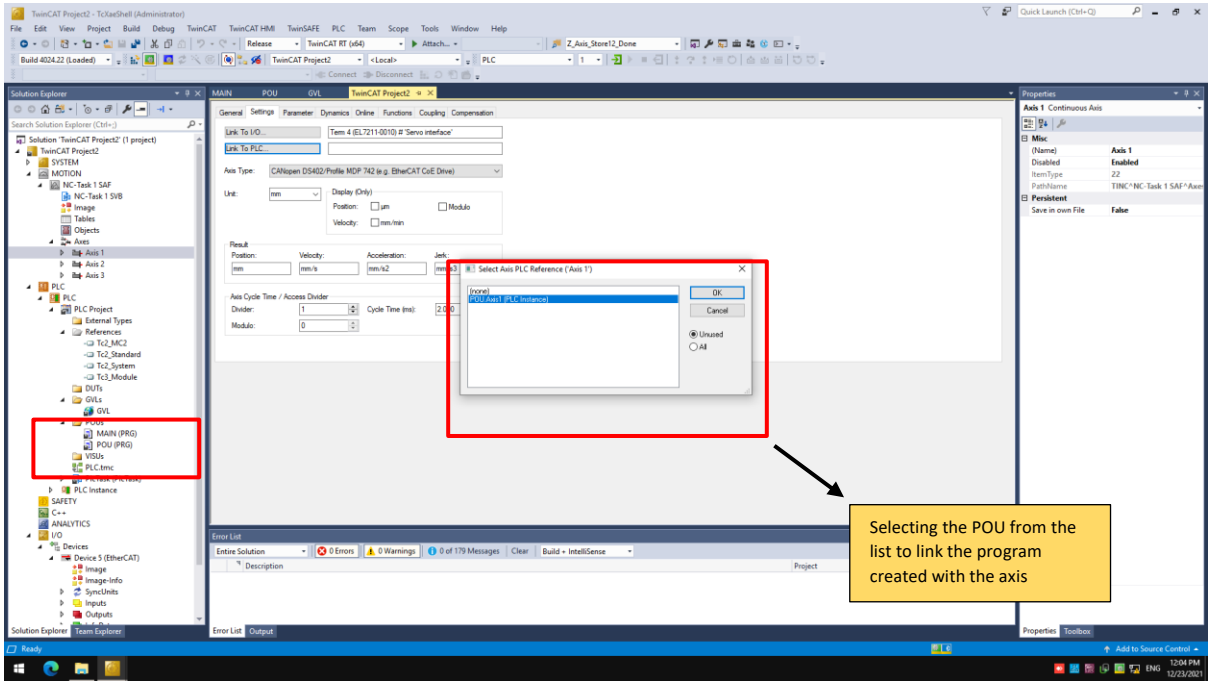

Disabling all the 3 controllers for Switching OFF the Motor – (Controller, Feed Fw, Feed Bw)


Selecting Operation types for various modes like – Absolute, Relative, Jog +, Jog -

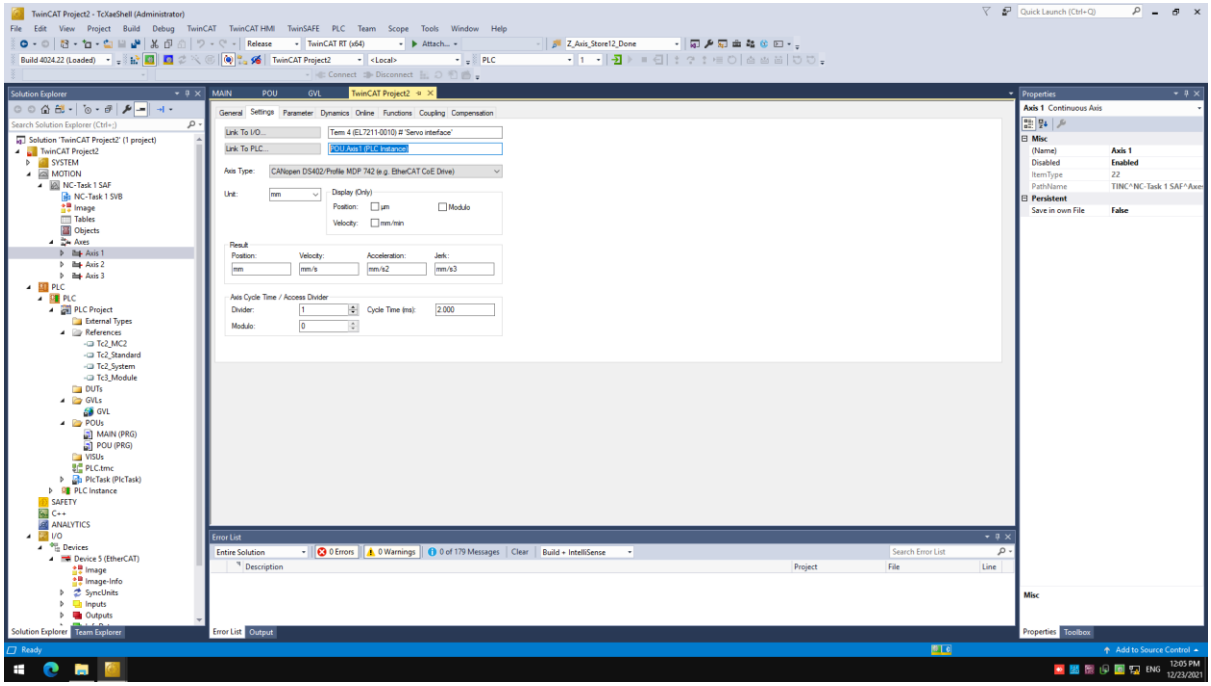

Select Settings for Linking the Motors to PLC program created

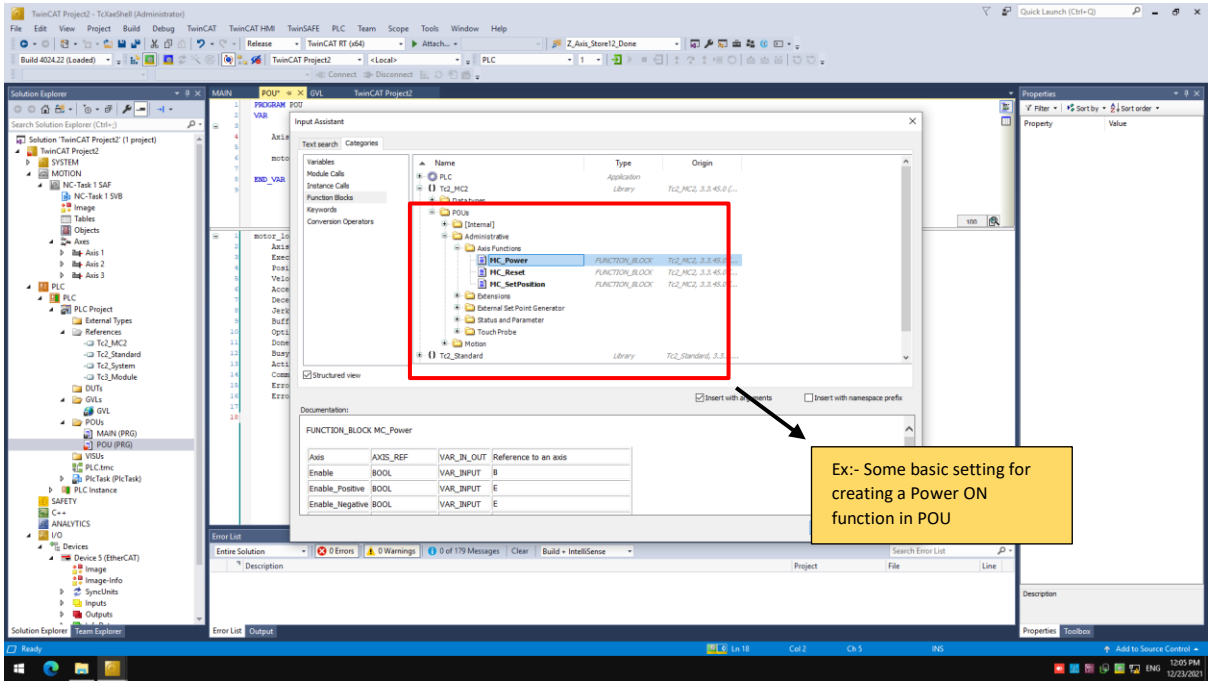
4.1.7 Linking Motors with PLC program

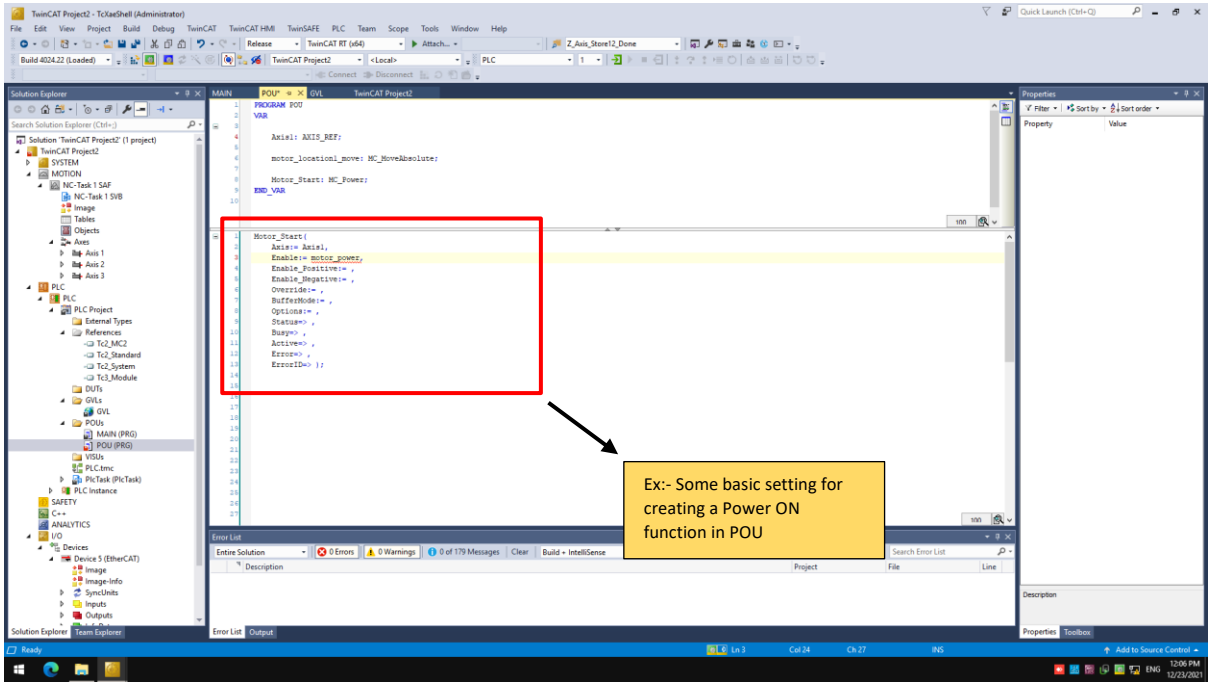


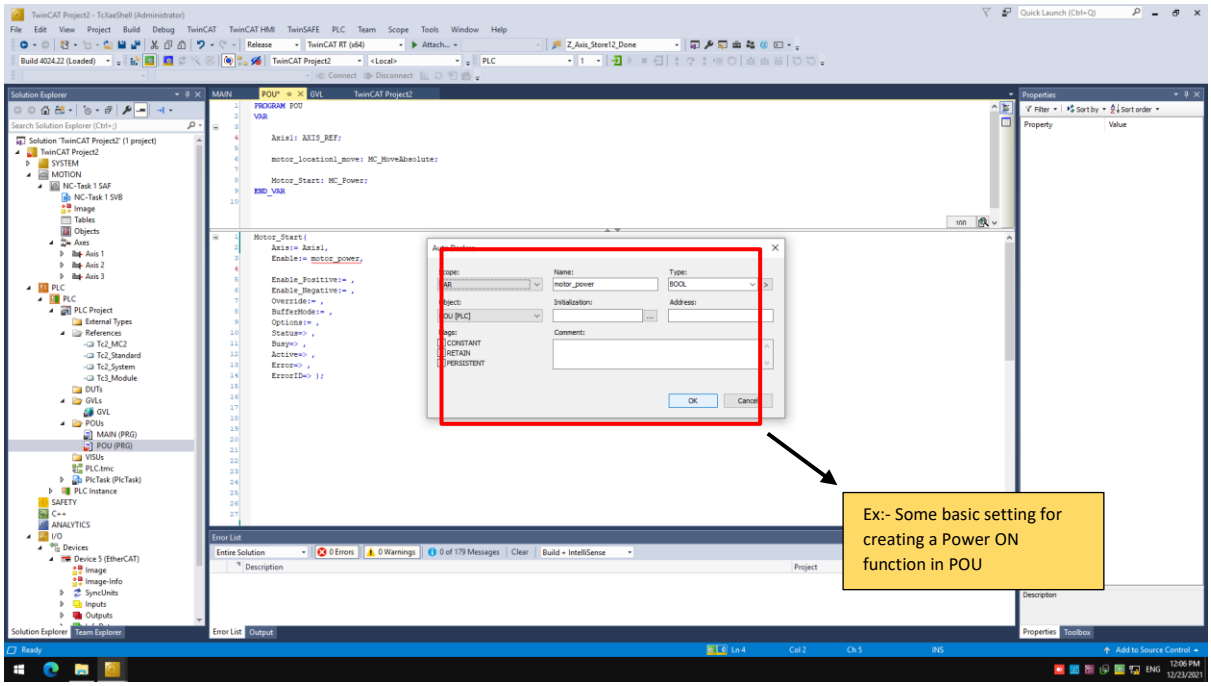


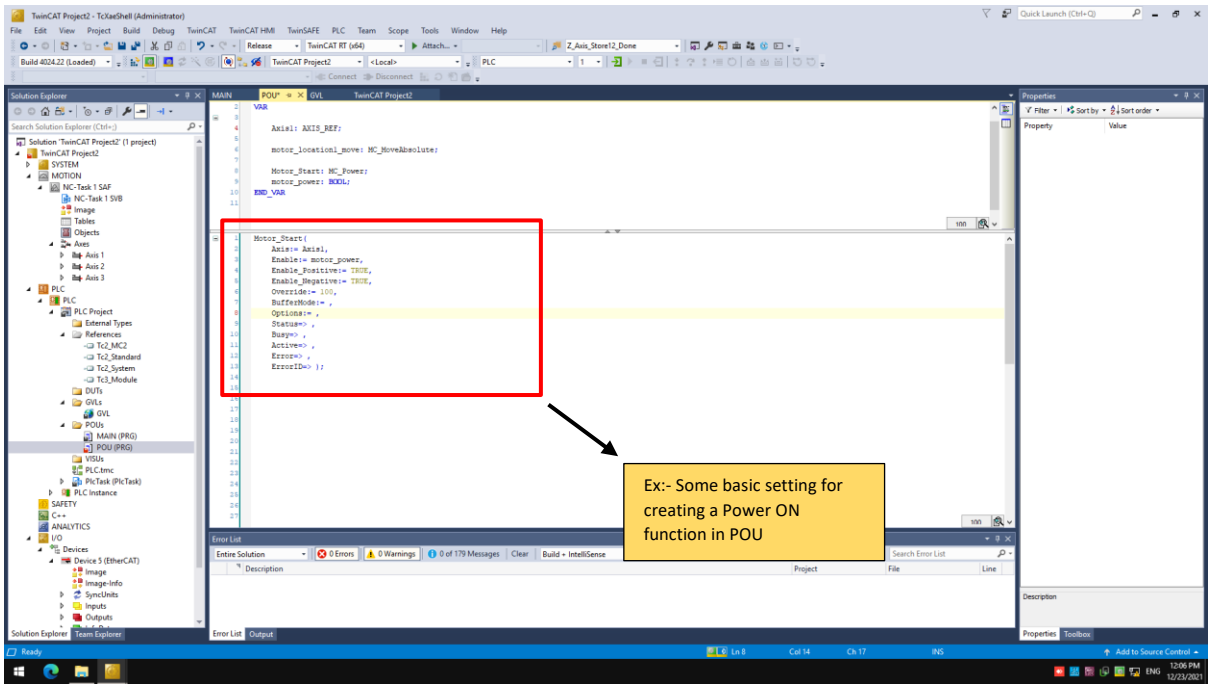
4.1.7.1 Example motor functions

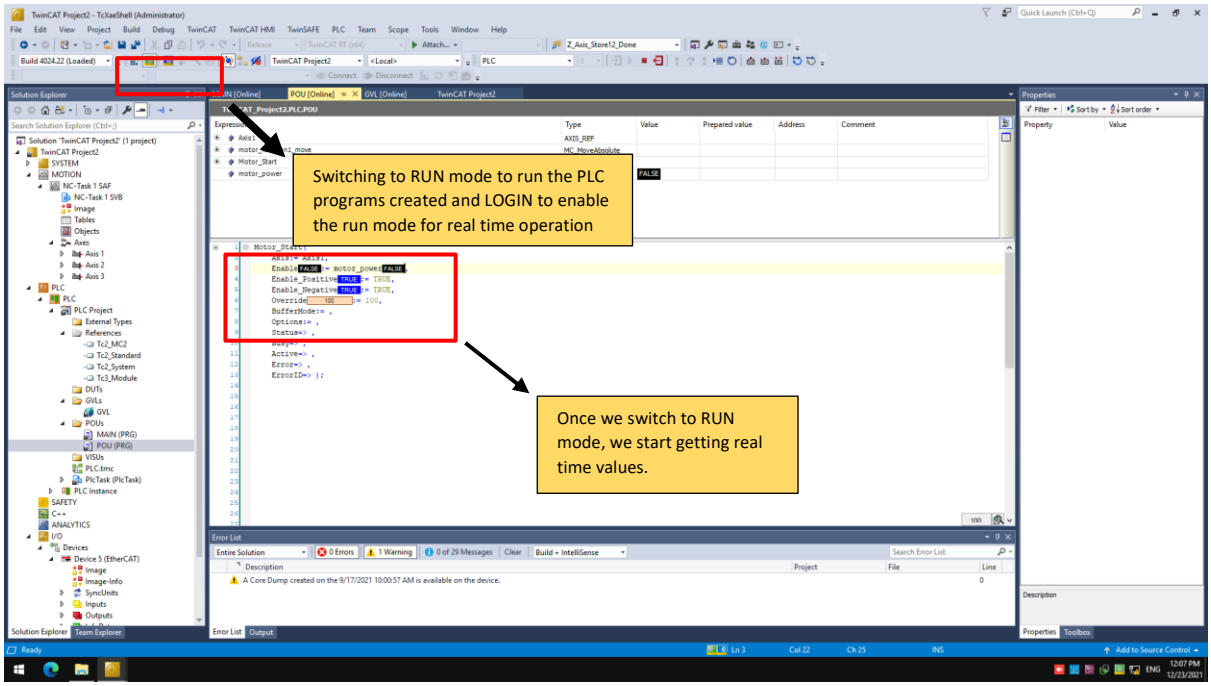


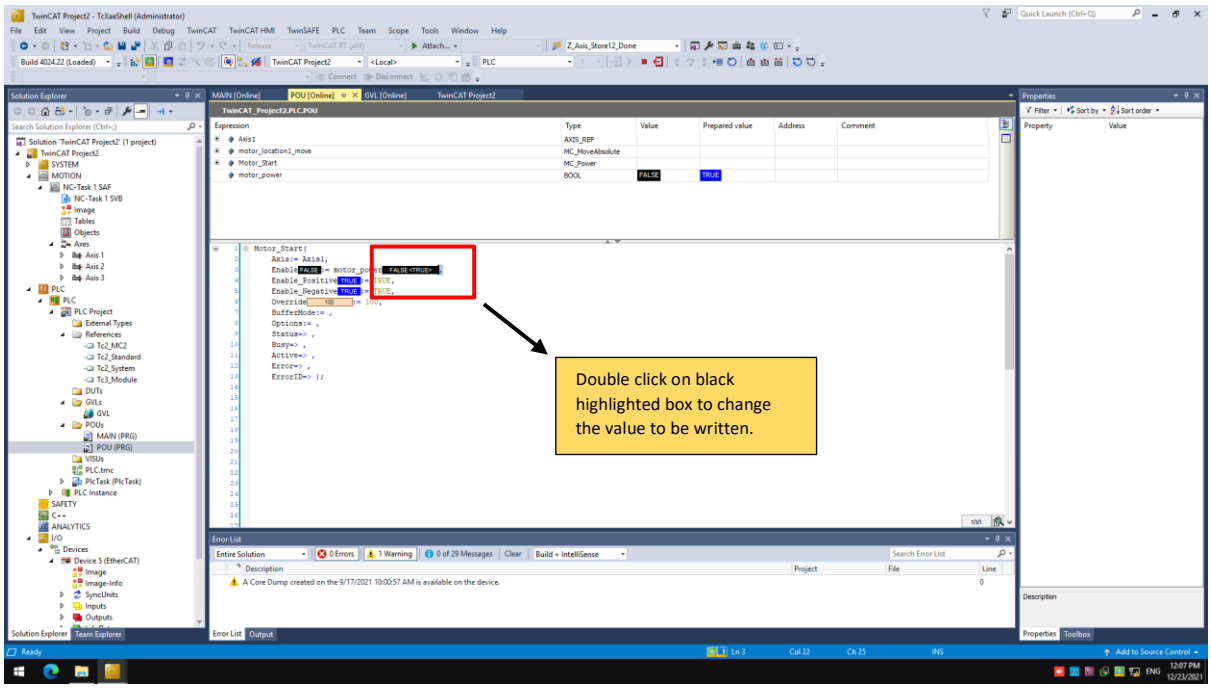


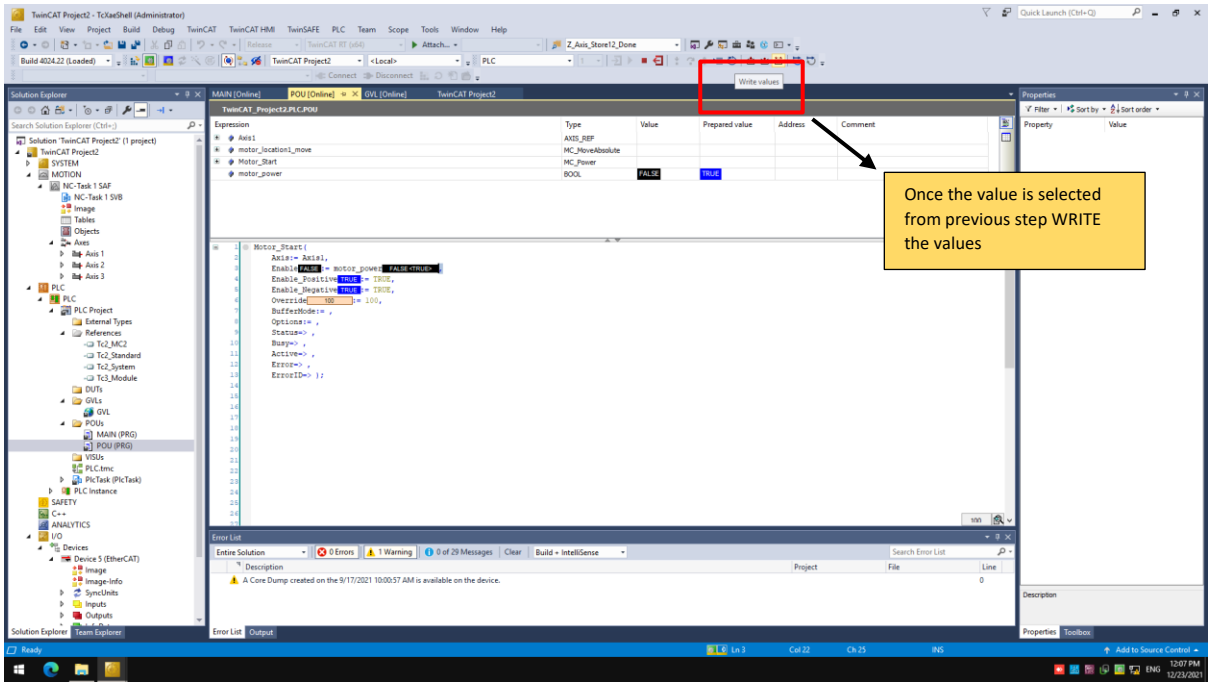

Ex:- Some basic setting for creating a Power ON function in POU

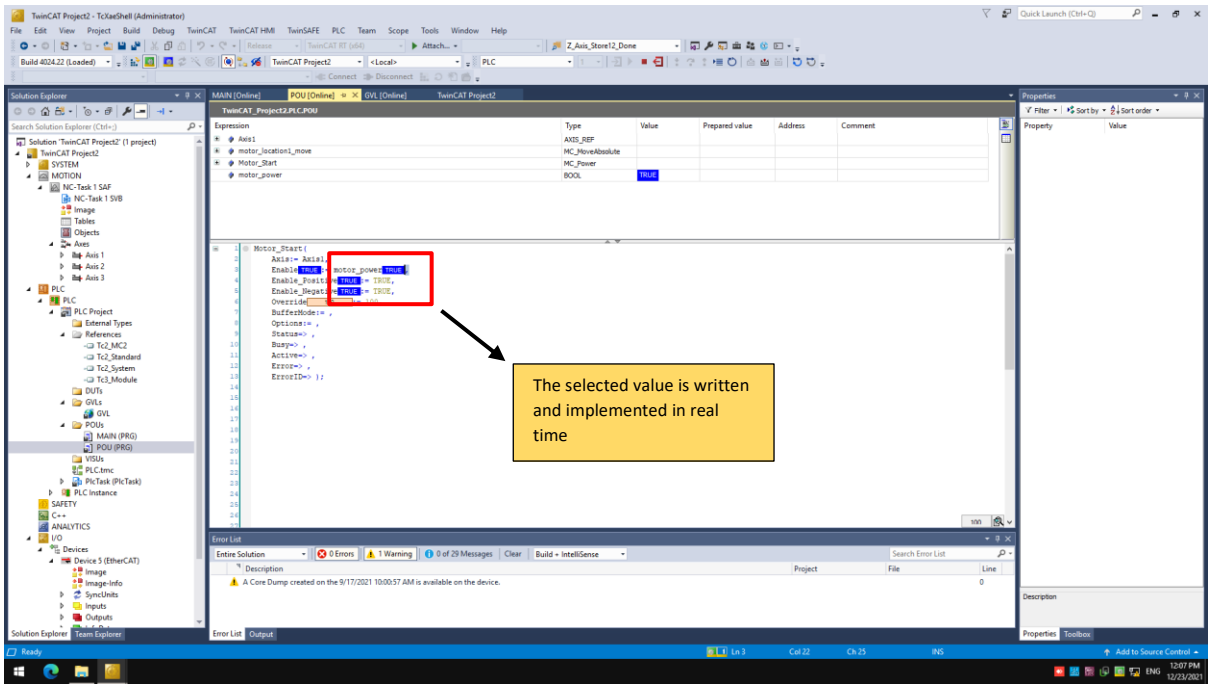

Ex:- Some basic setting for creating a Power ON function in POU

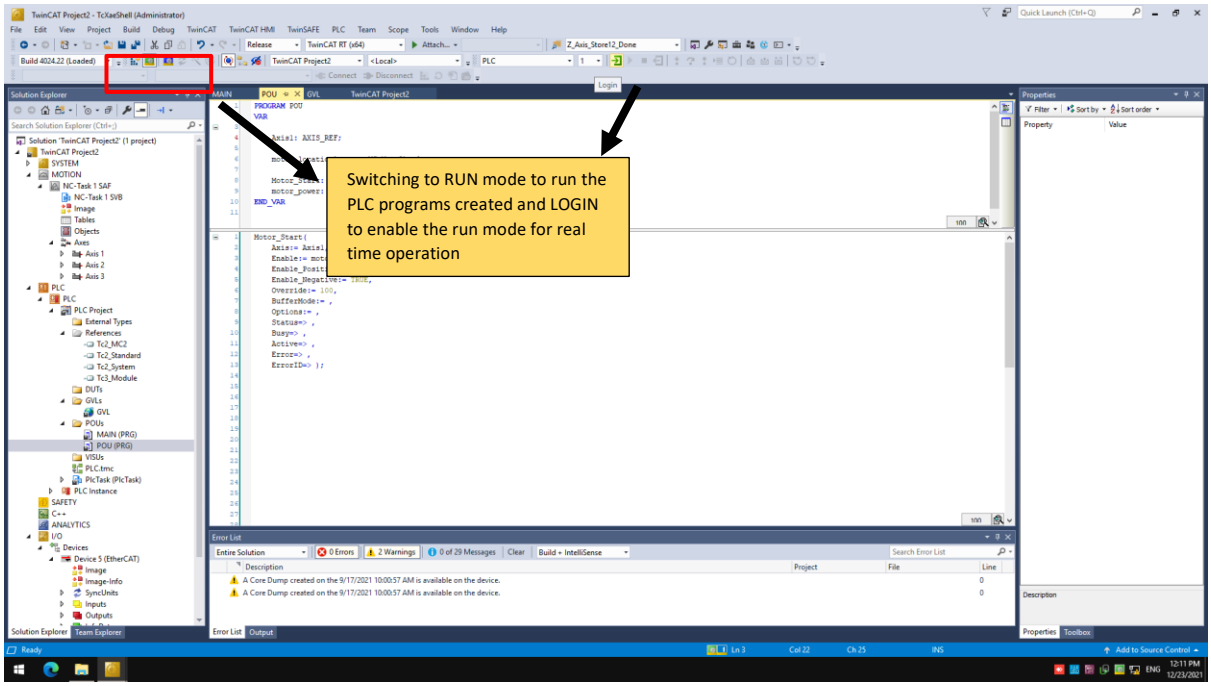

Ex:- Some basic setting for creating a Power ON function in POU


Switching to RUN mode to run the PLC programs created and LOGIN to enable the run mode for real time operation
Once we switch to RUN mode, we start getting real time values.

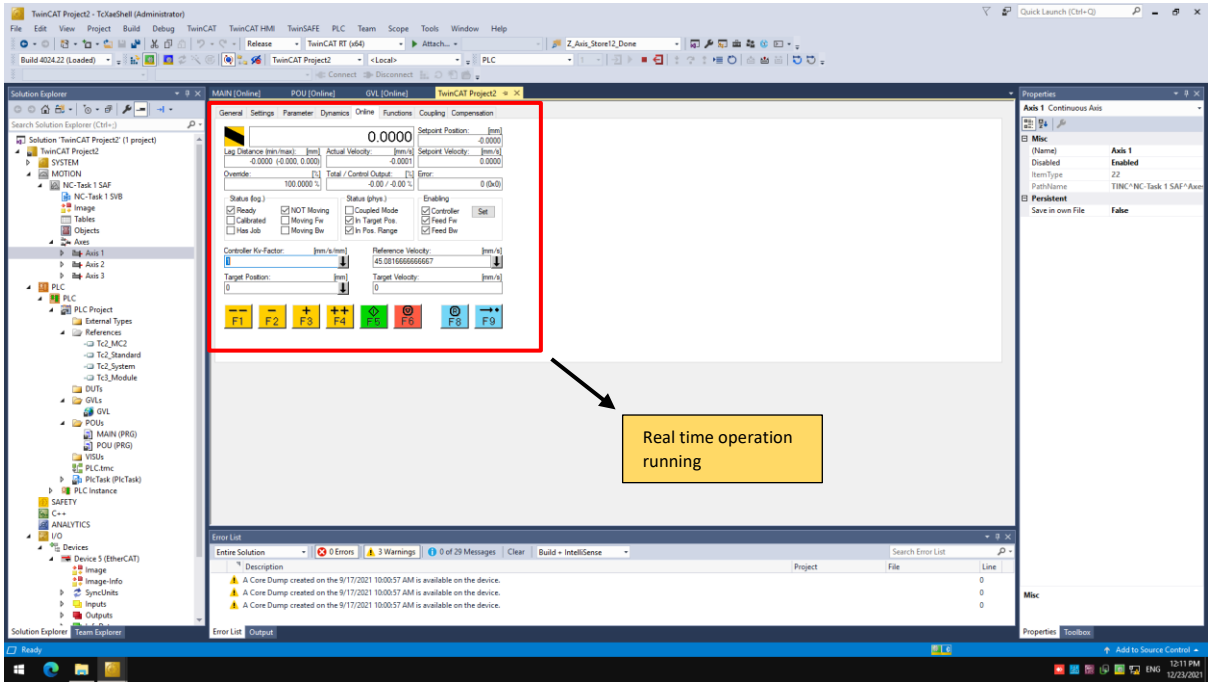

Double click on black highlighted box to change the value to be written.

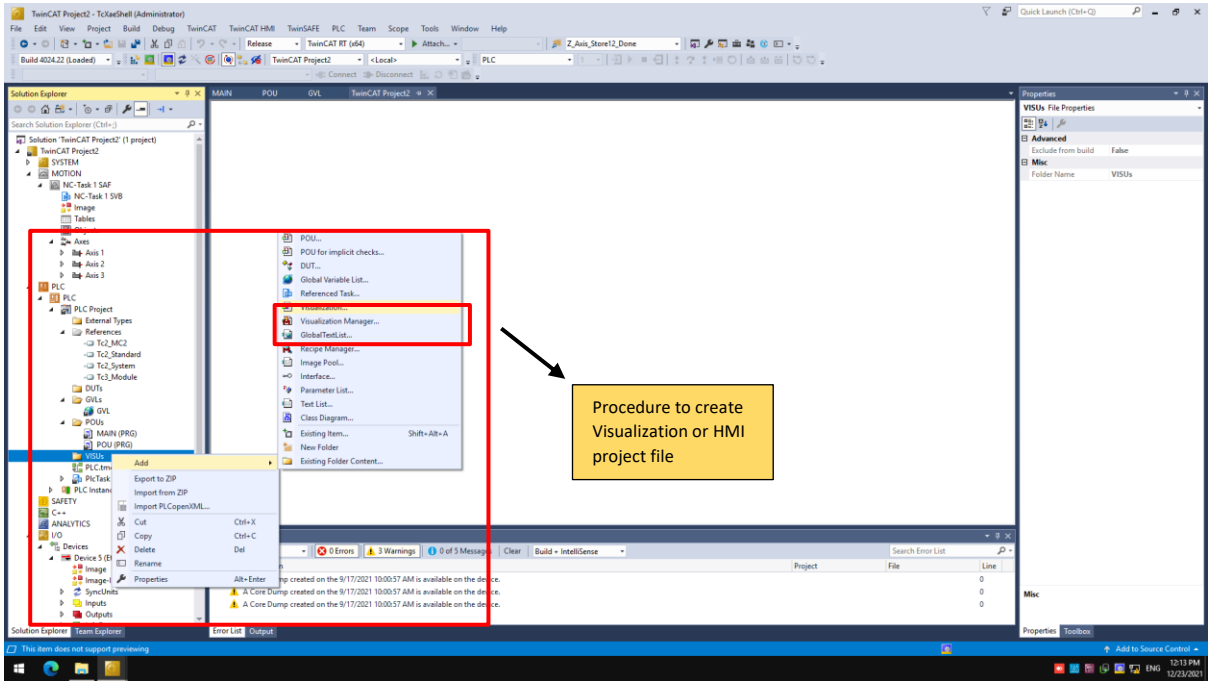

Write values
Once the value is selected from previous step WRITE the values

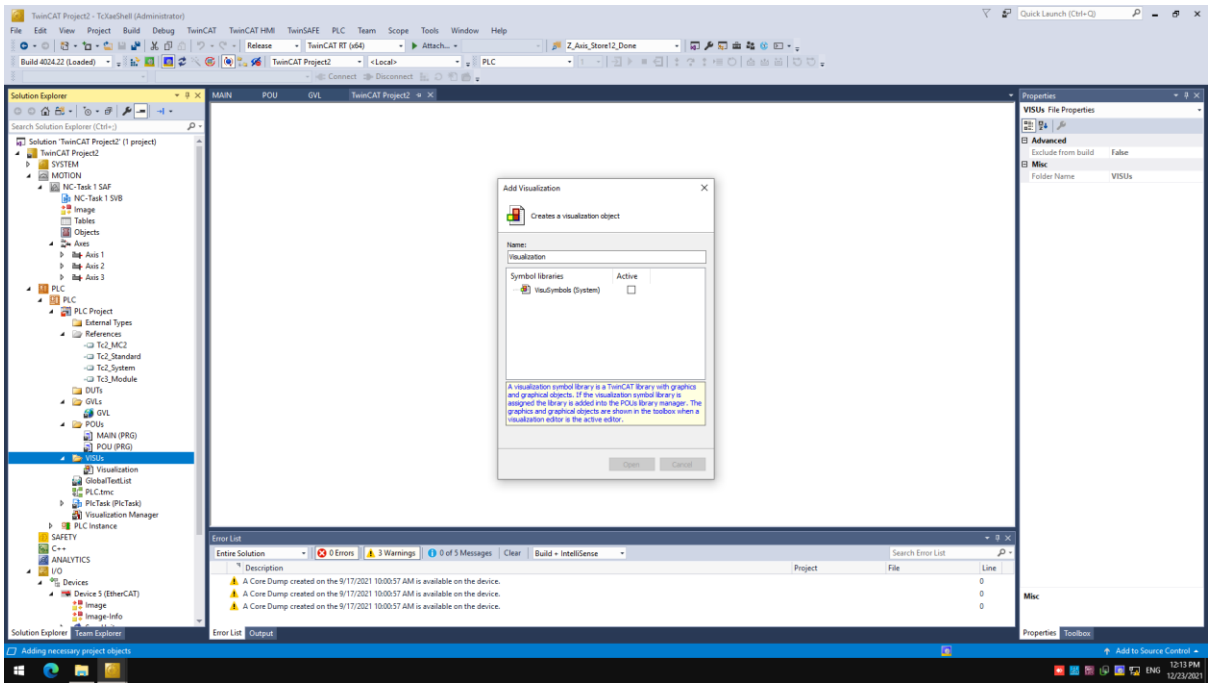

The selected value is written and implemented in real time

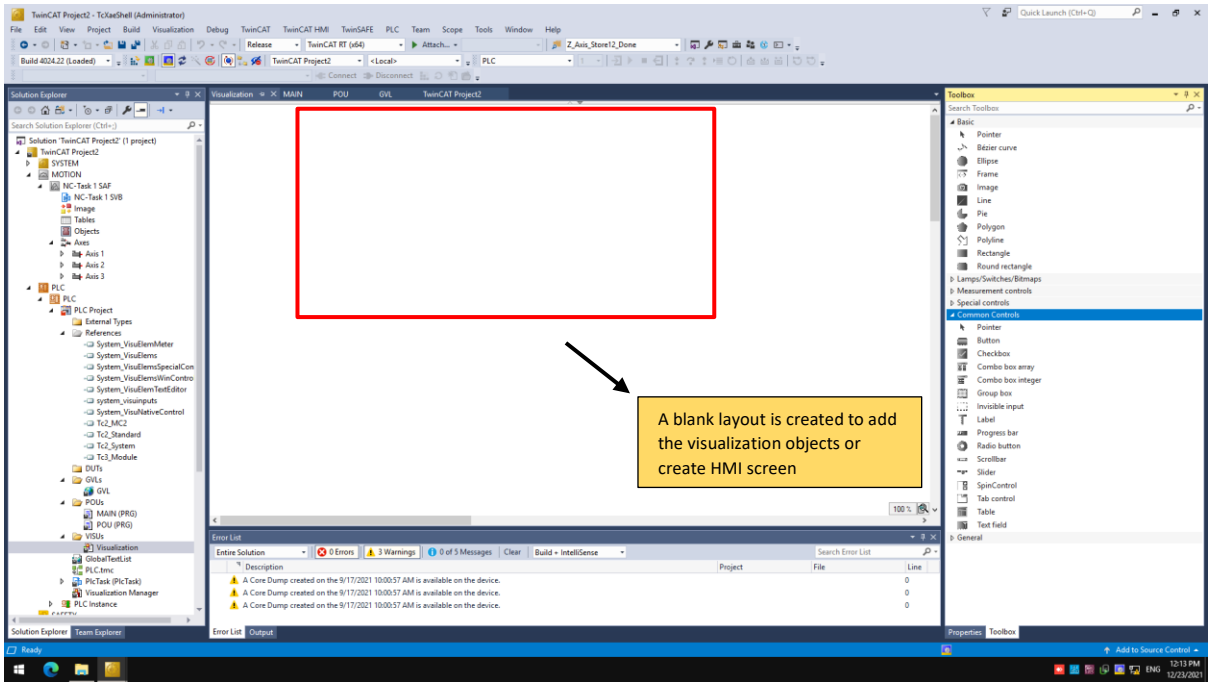

Switching to RUN mode to run the PLC programs created and LOGIN to enable the run mode for real time operation

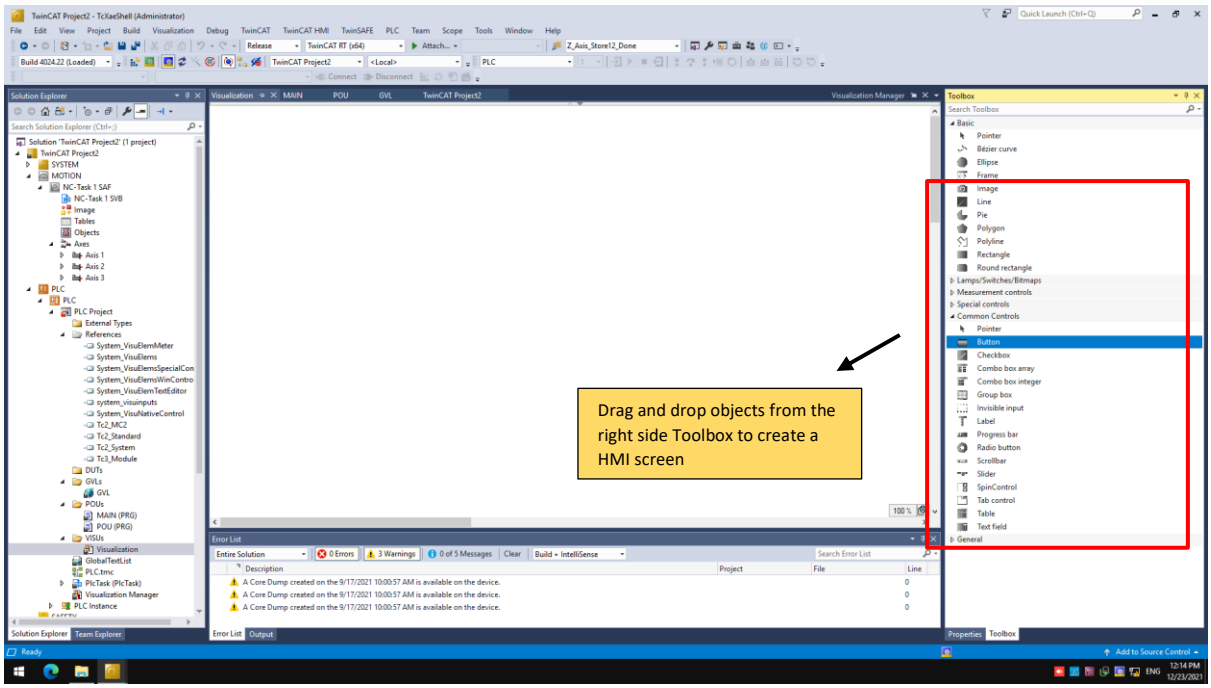
4.1.8 Creating Visualisation or HMI screen

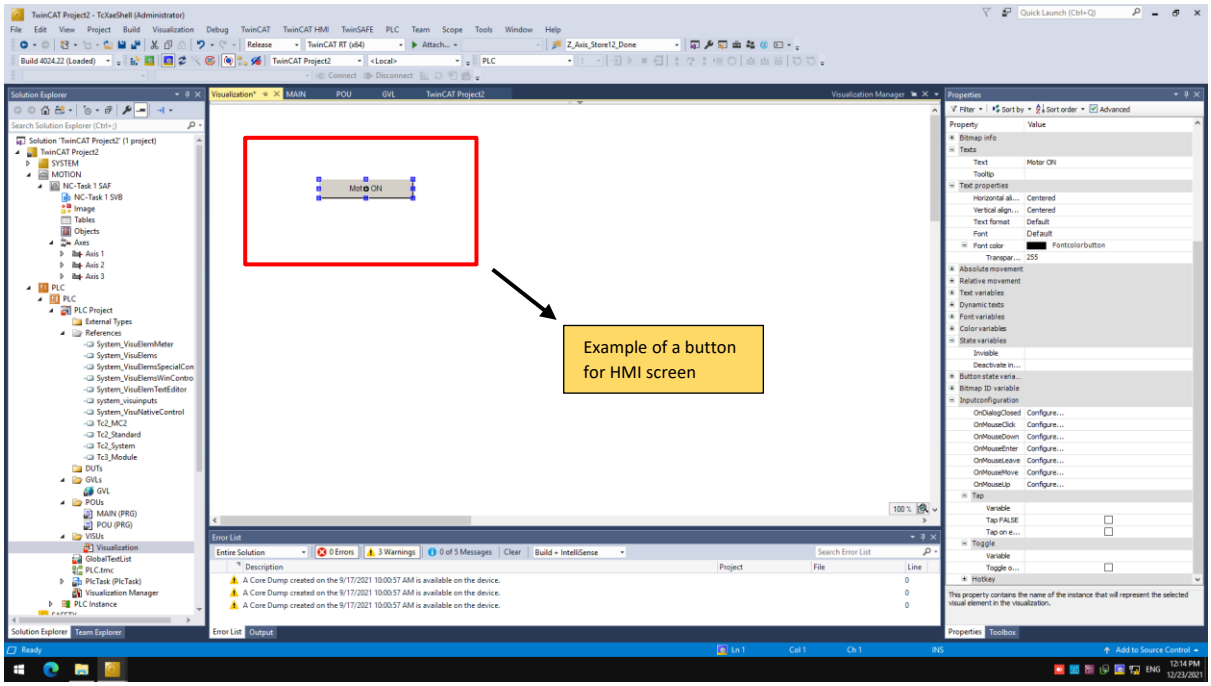


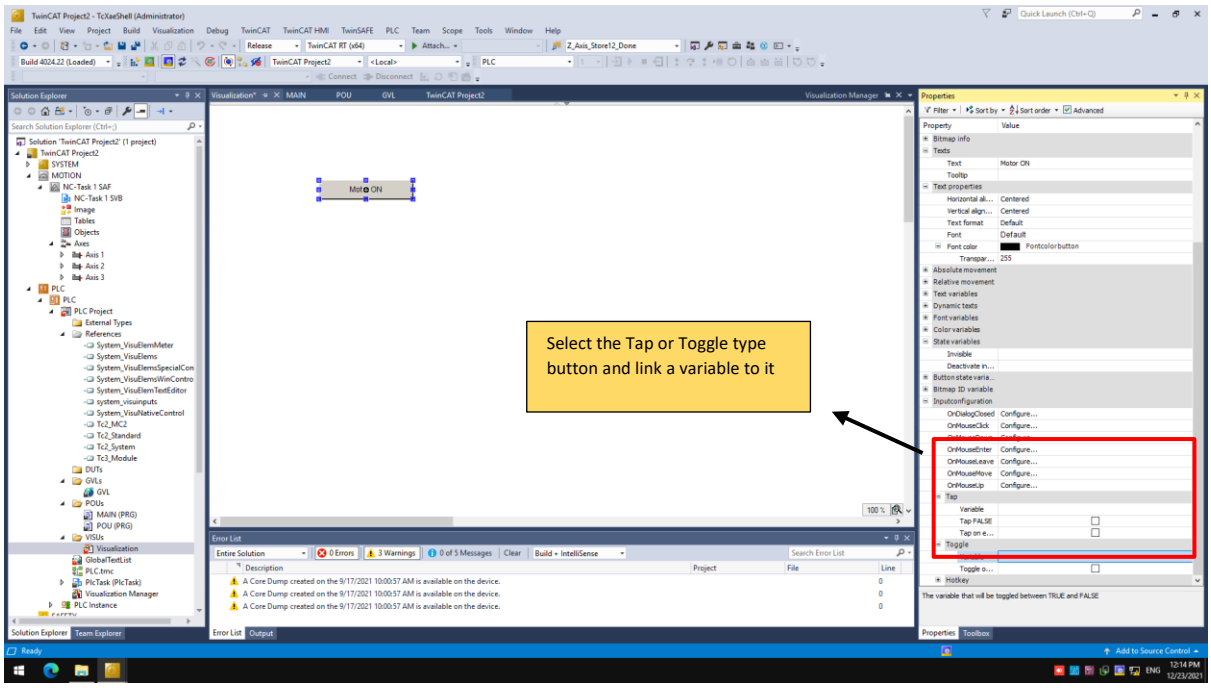


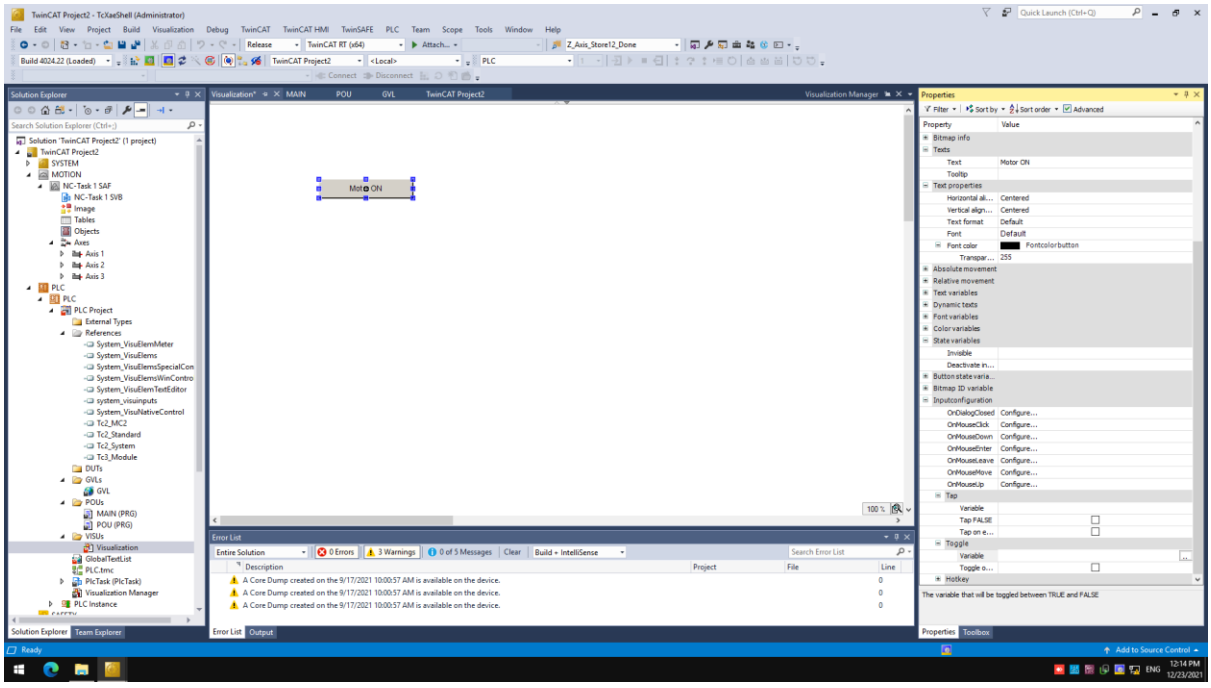

Add Visualization
Creates a visualization object
Name:
Visualization
Symbol libraries
Active
VisuSymbols (System)

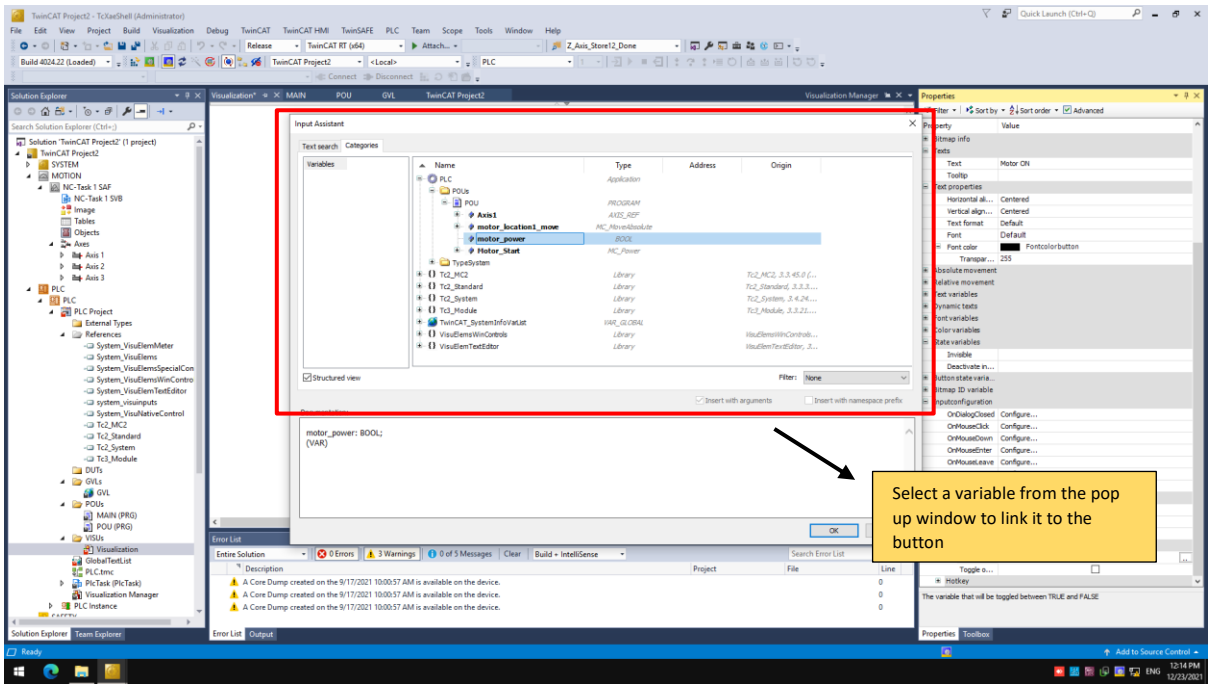

A blank layout is created to add the visualization objects or create HMI screen

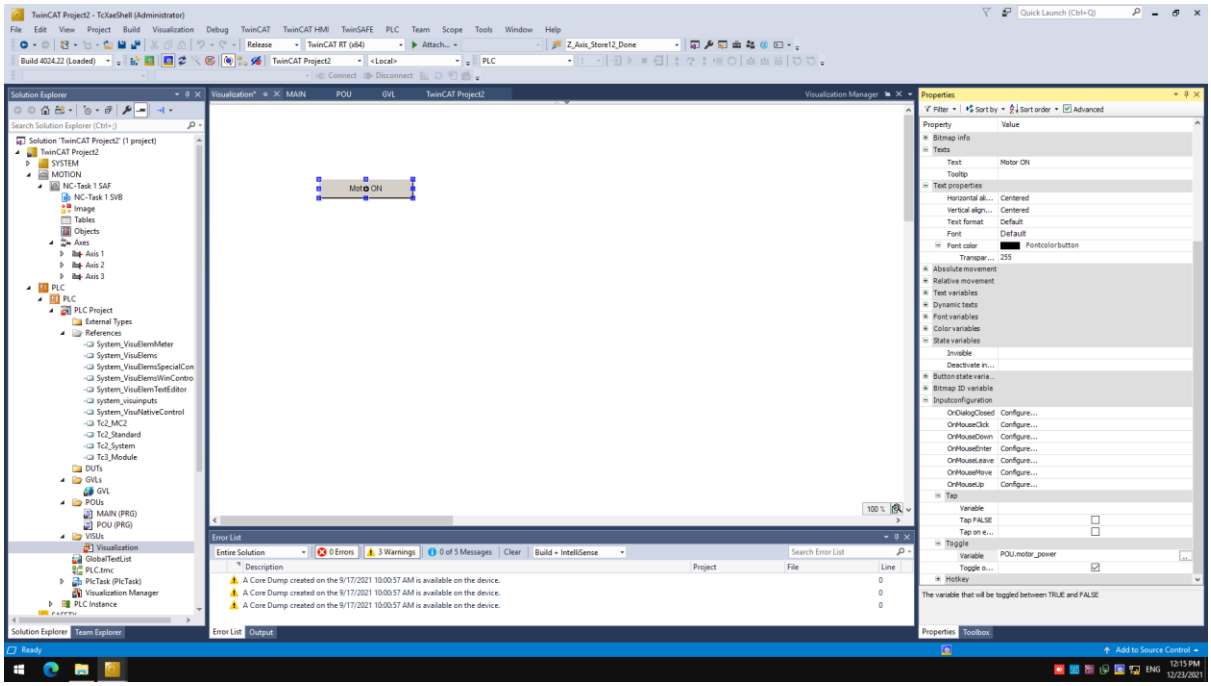

Drag and drop objects from the right side Toolbox to create a HMI screen


Motor ON
Example of a button for HMI screen

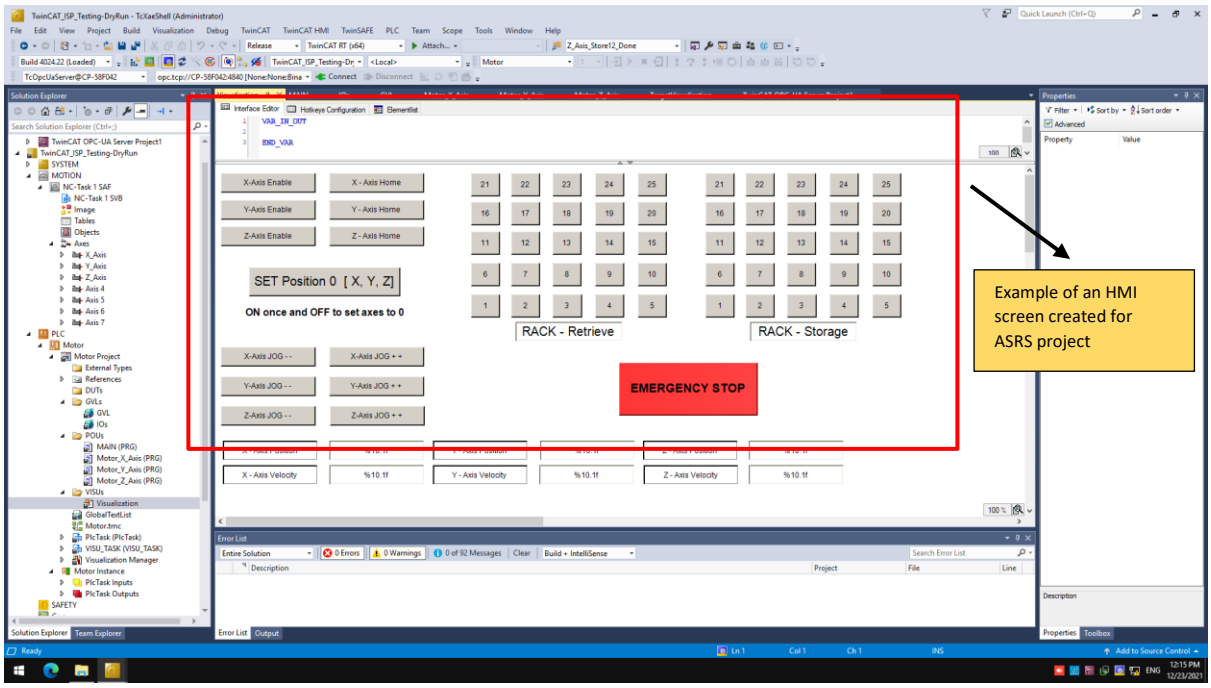

Select the Tap or Toggle type button and link a variable to it

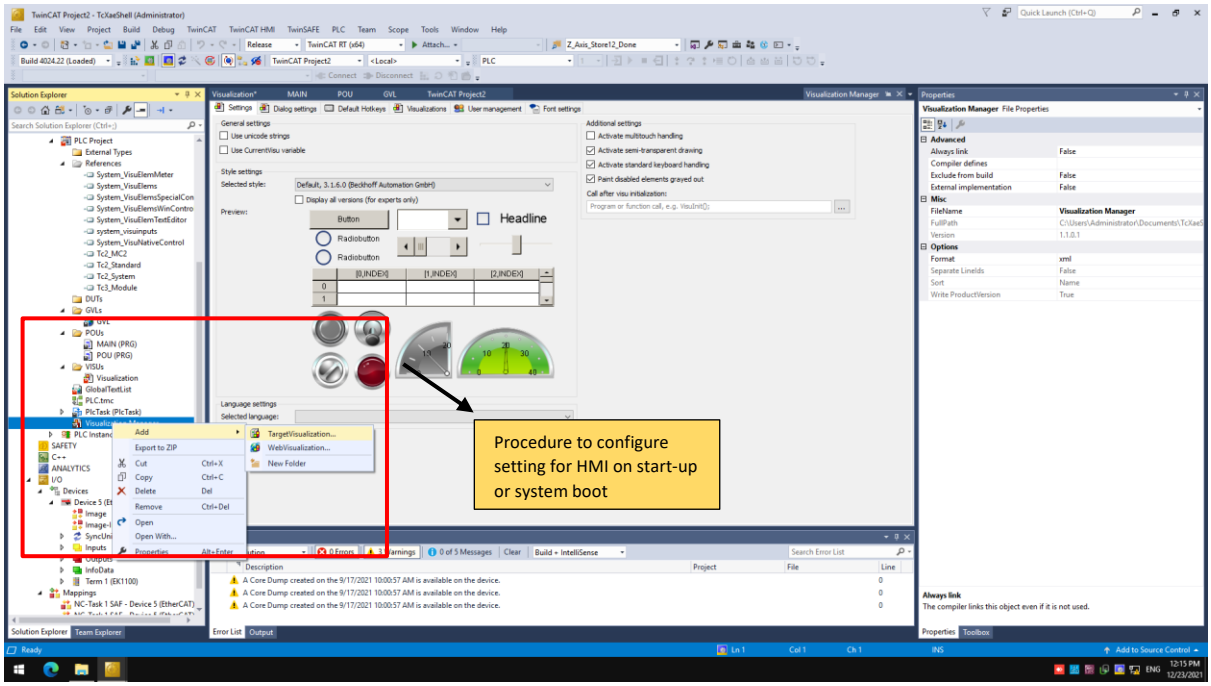


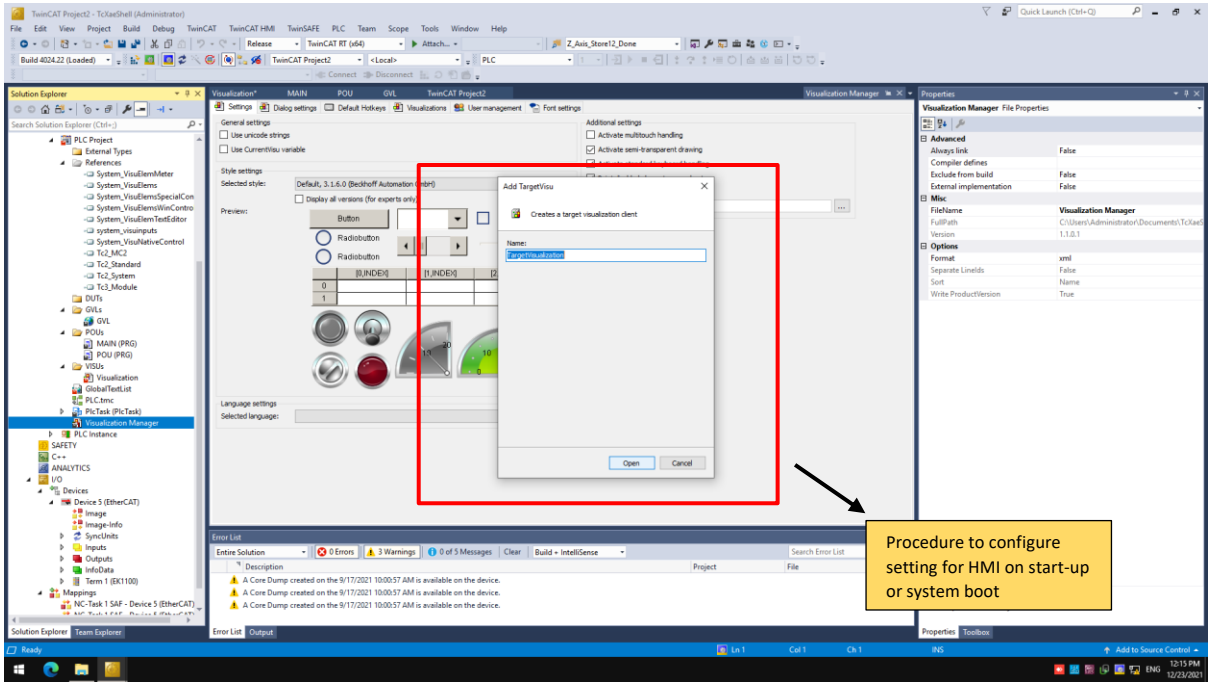

Select a variable from the pop up window to link it to the button

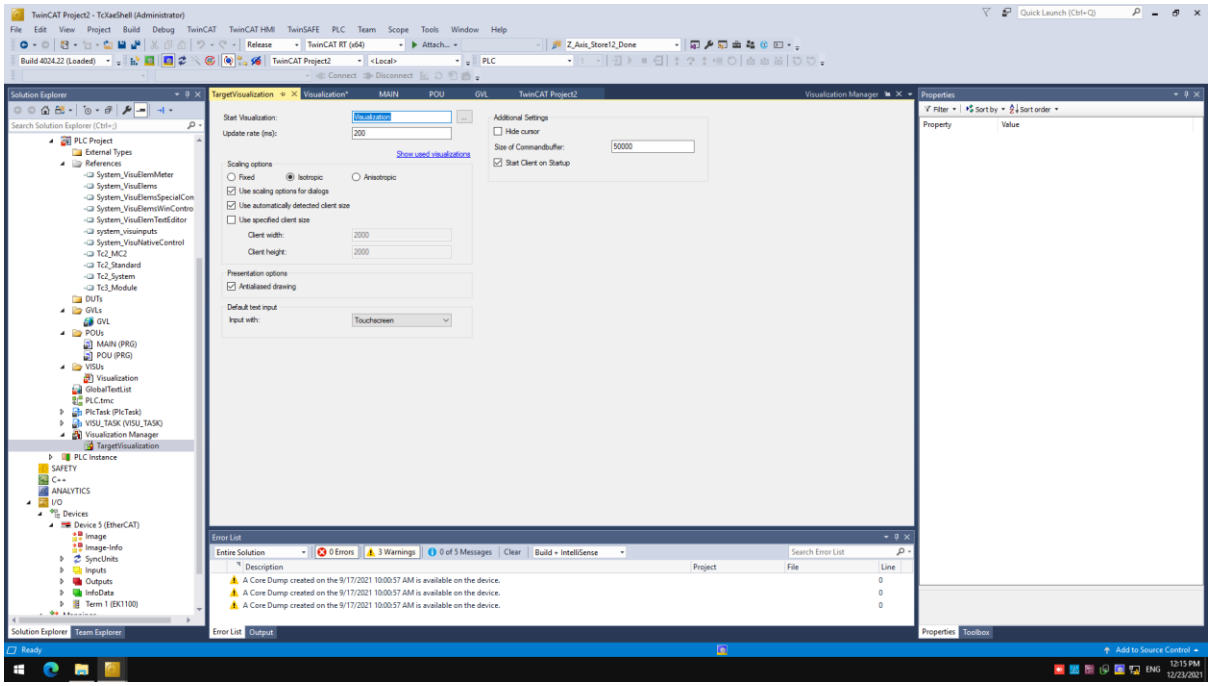


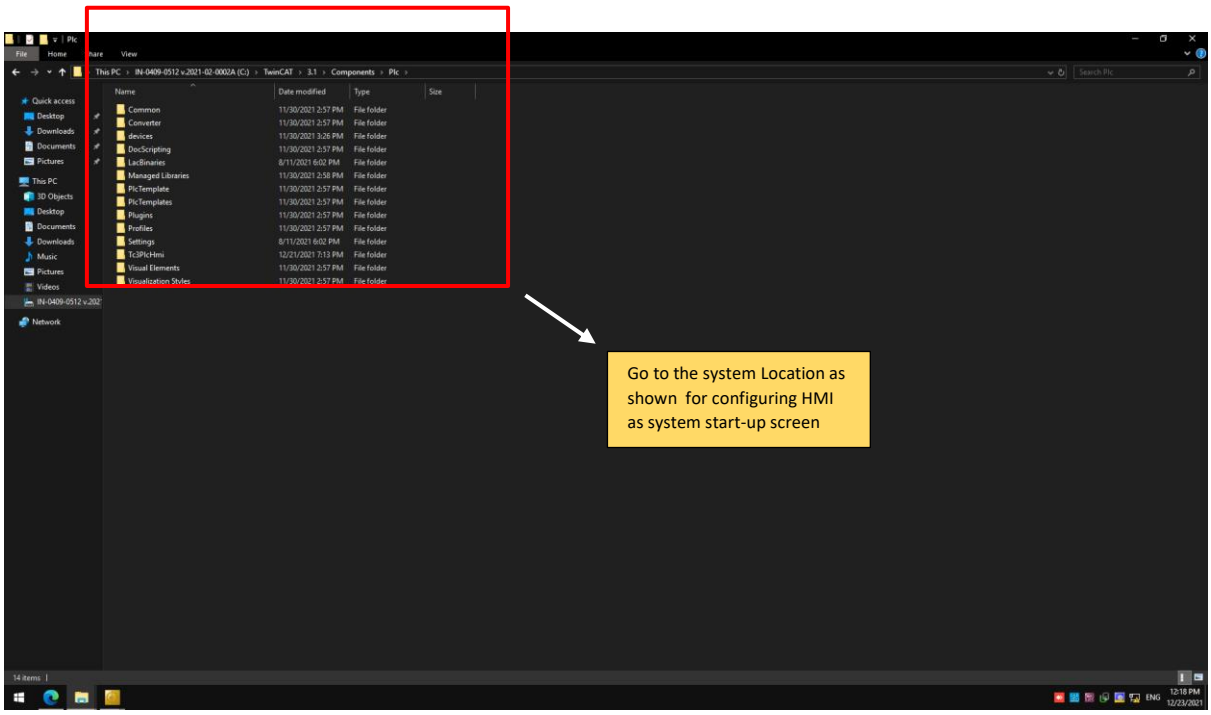
4.1.8.1 HMI settings and system configuration for boot up

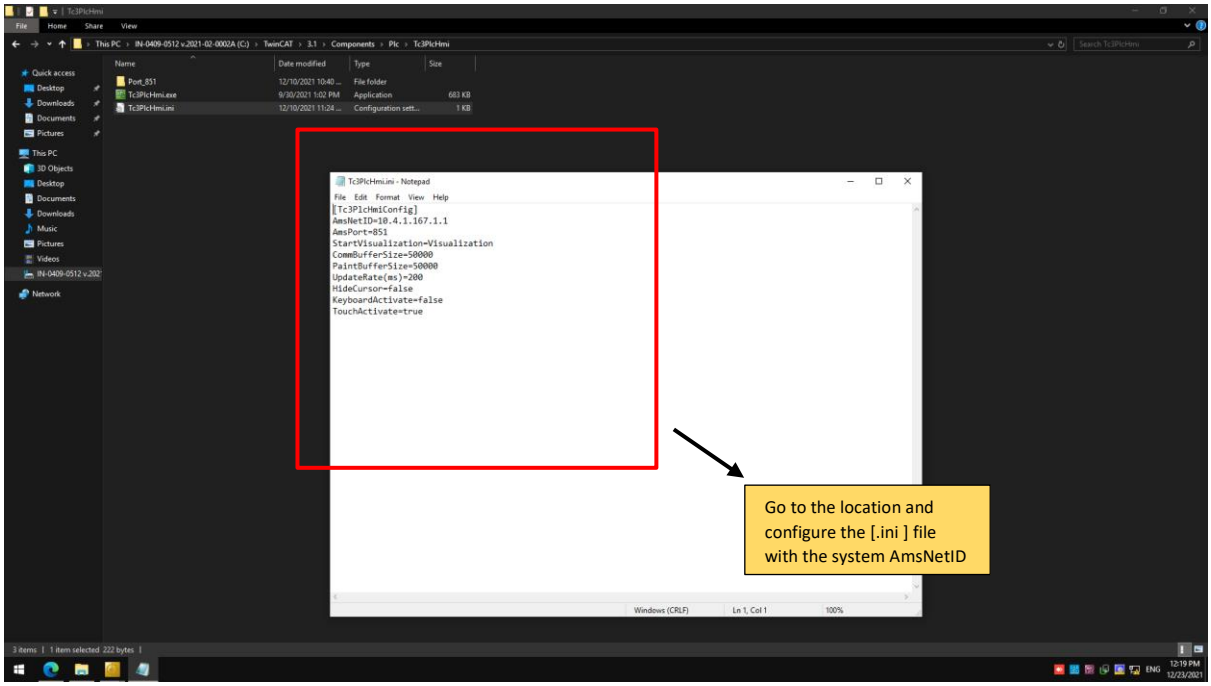




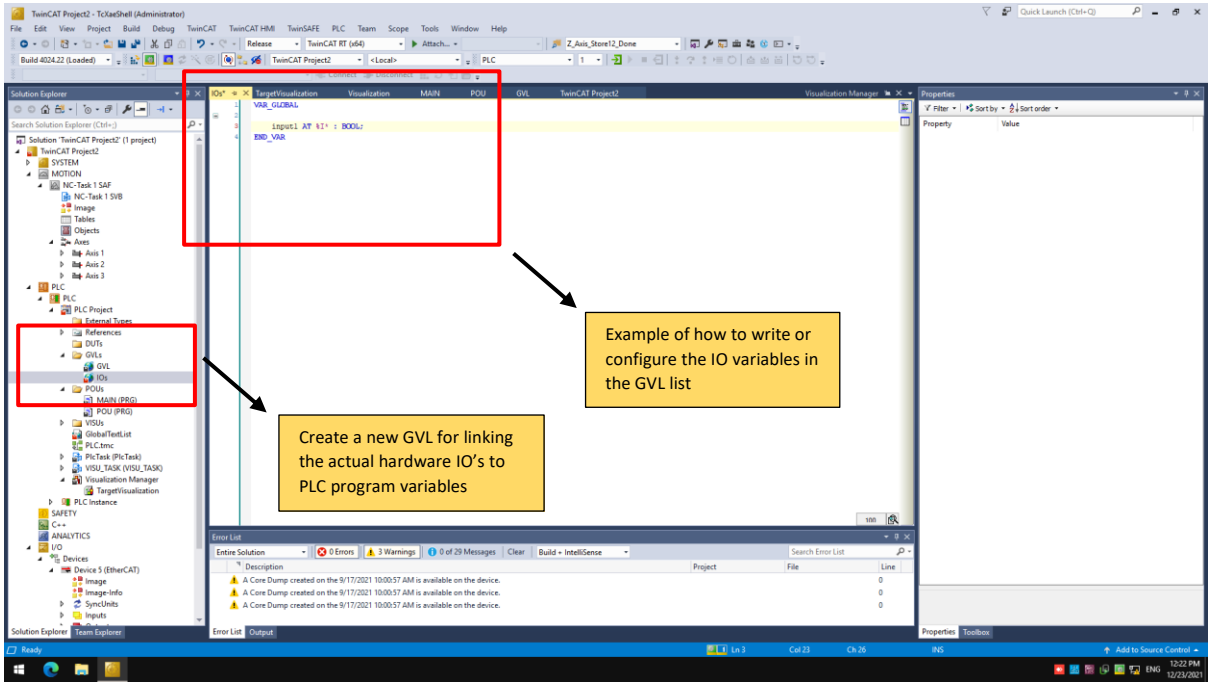

Add TargetVisu
Creates a target visualization client
Name:
TargetVisualization
Open
Cancel
Procedure to configure setting for HMI on start-up or system boot

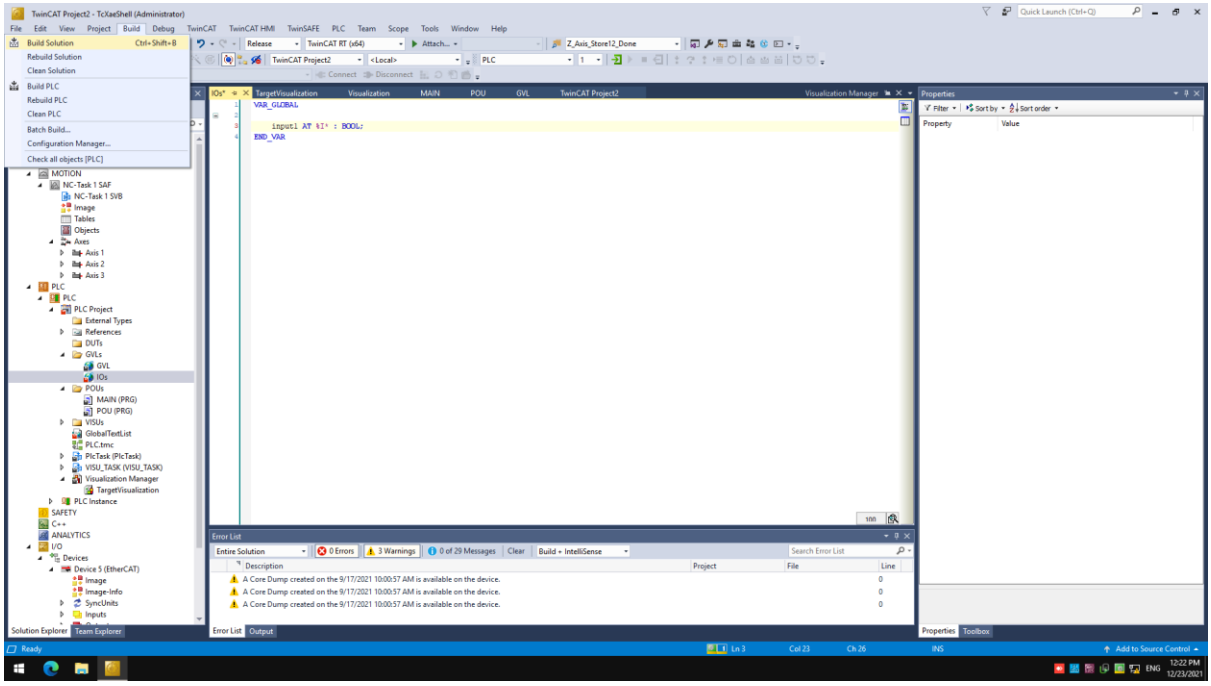


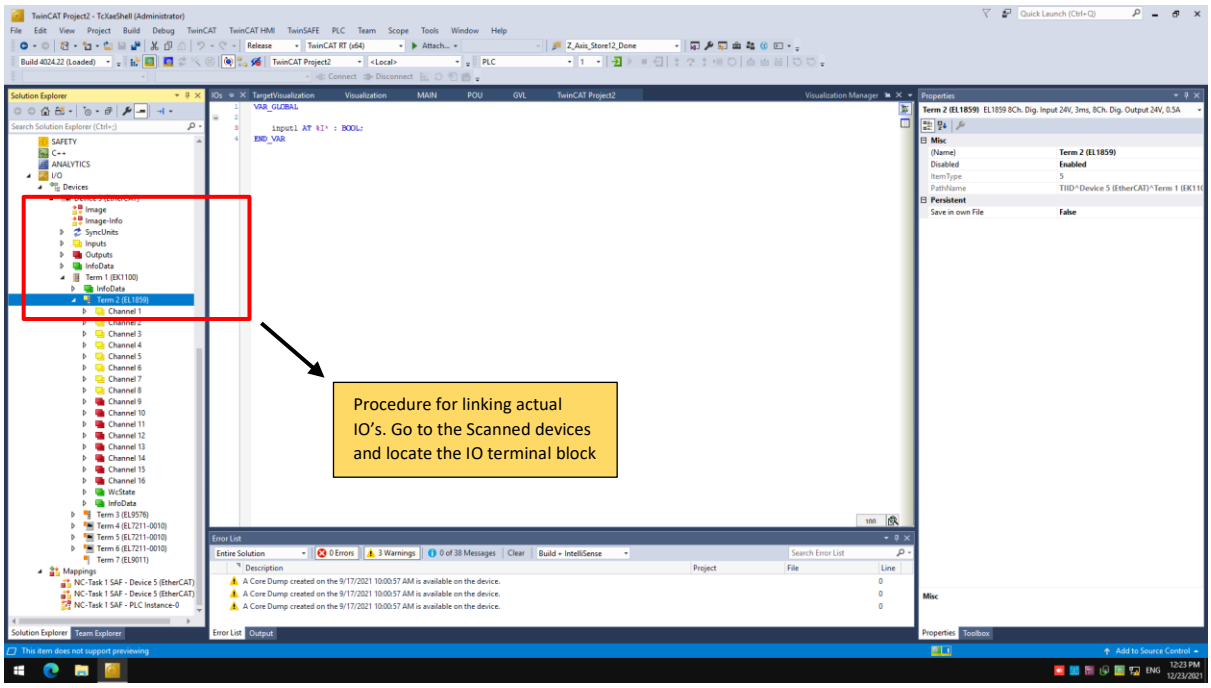

Go to the system Location as shown for configuring HMI as system start-up screen

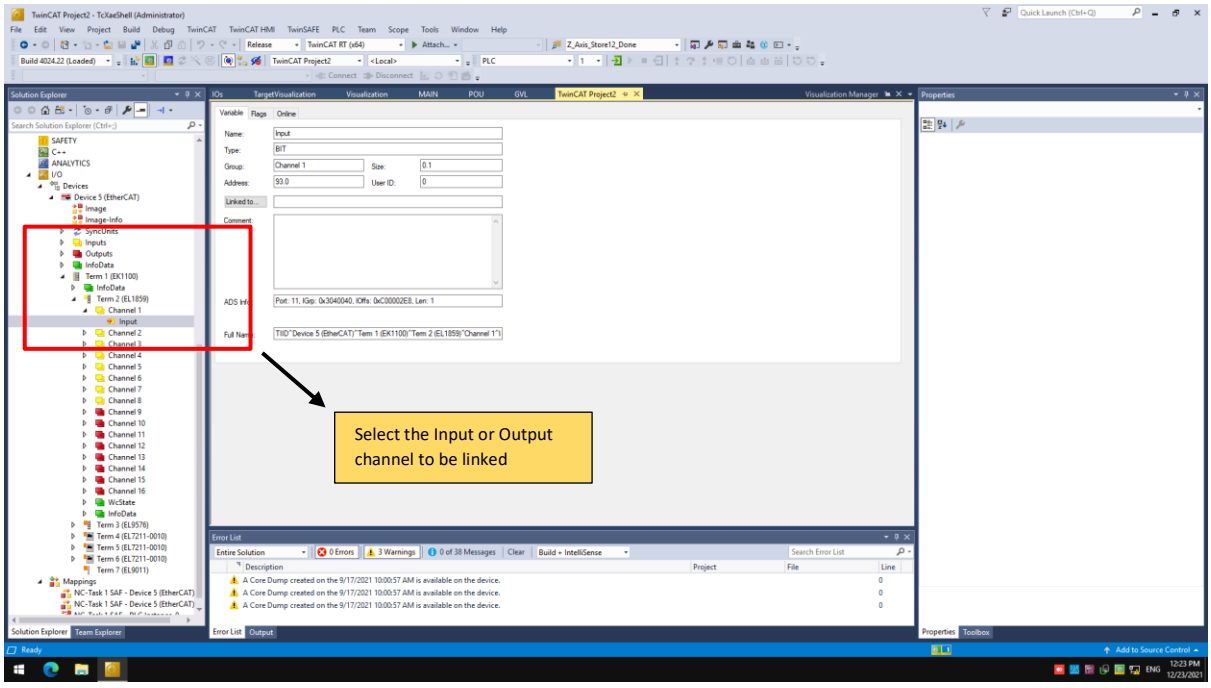

Tc3PlcHmi.ini - Notepad
[Tc3PlcHmiConfig]
AmsNetID=10.4.1.167.1.1
AmsPort=851
StartVisualization=Visualization
CommBufferSize=50000
PaintBufferSize=500000
UpdateRate(ms)=200
HideCursor=false
KeyboardActivate=false
TouchActivate=true
Go to the location and configure the [.ini] file with the system AmsNetID

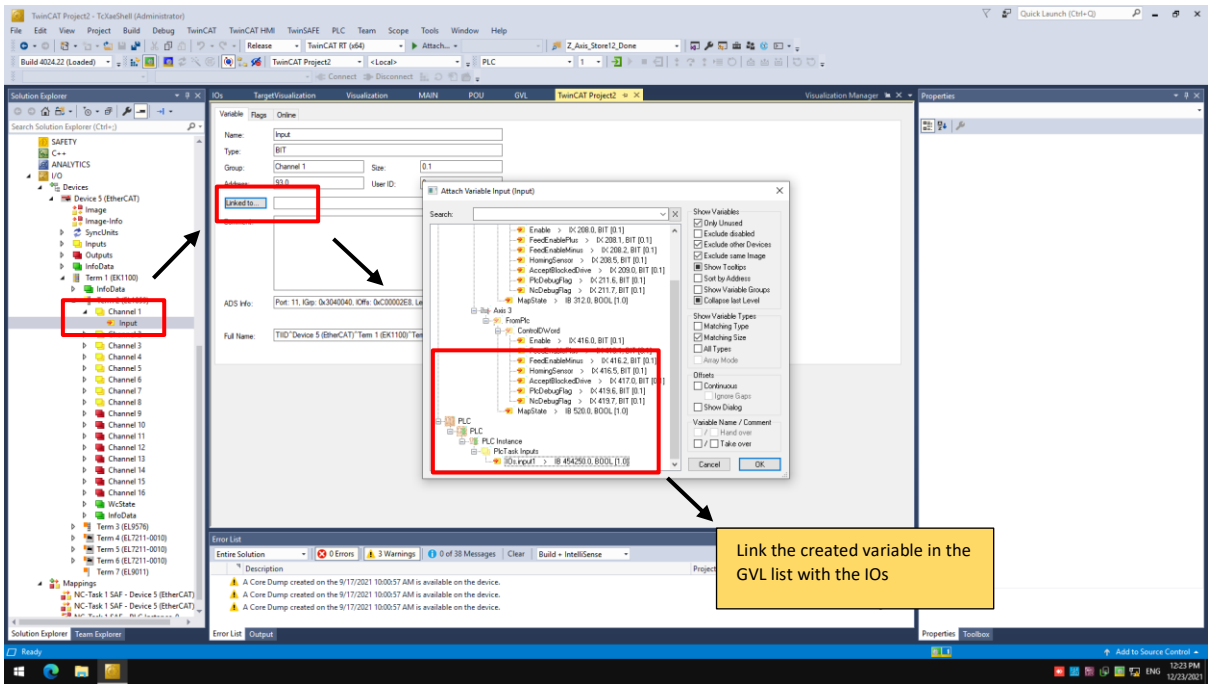
4.1.9 Examples Hardware IO's linking with PLC program

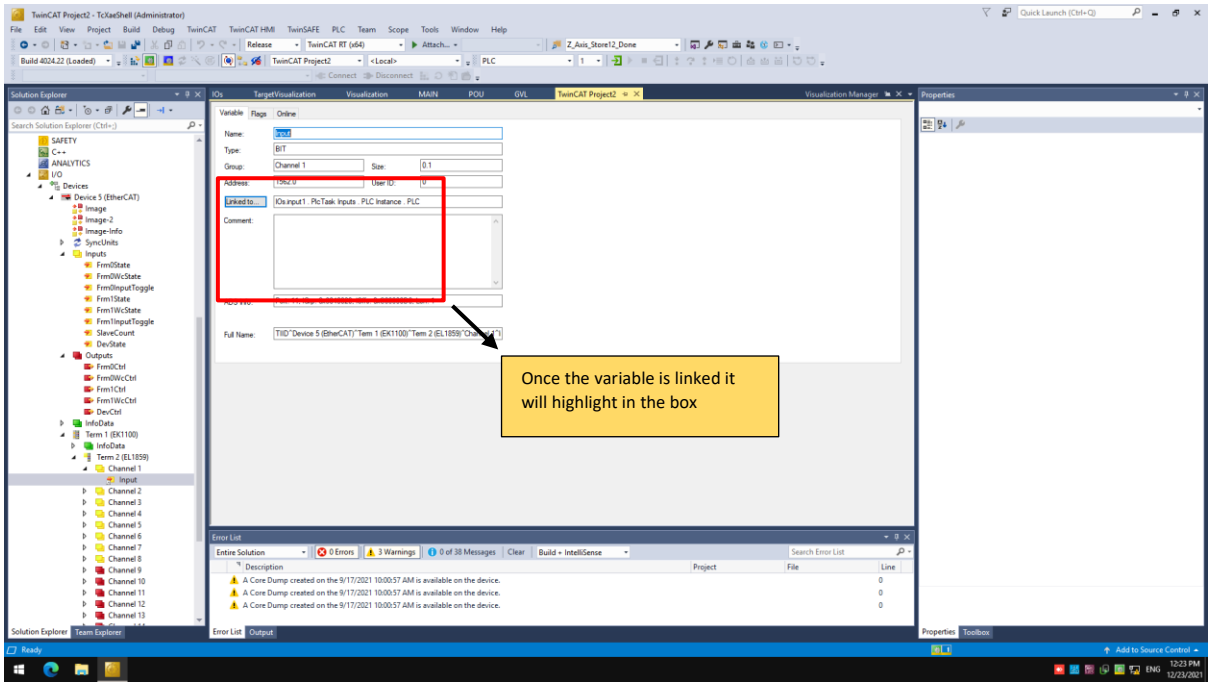




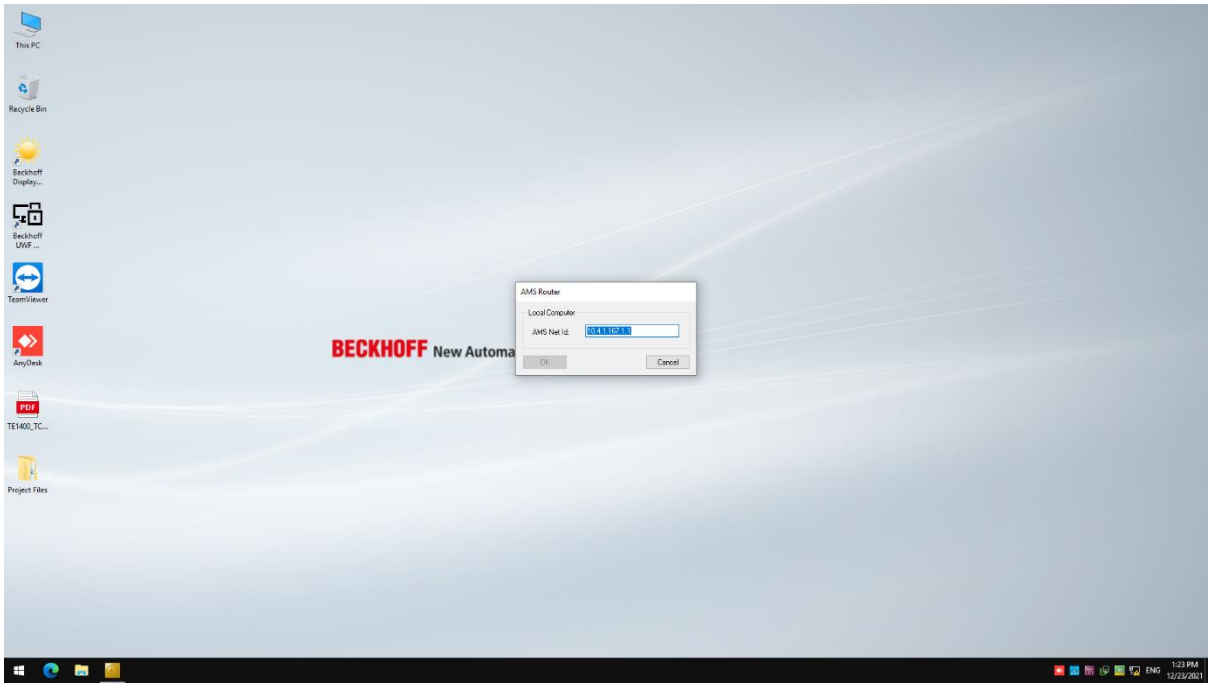

Procedure for linking actual IO's. Go to the Scanned devices and locate the IO terminal block

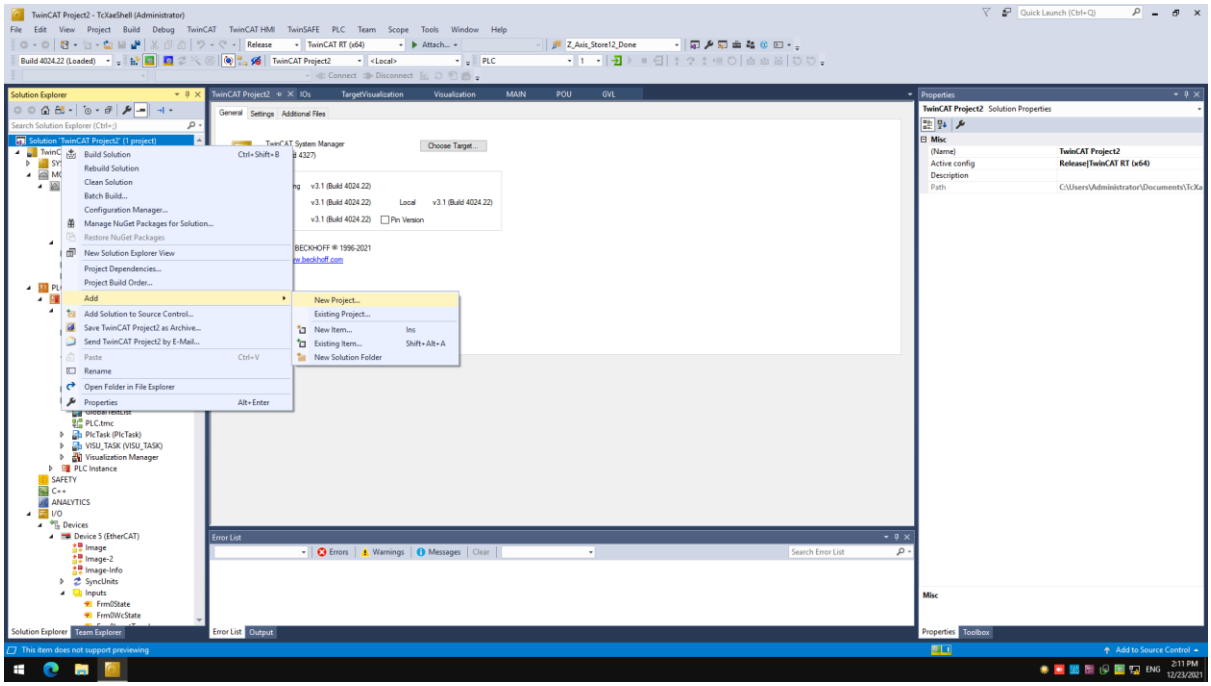

Select the Input or Output channel to be linked

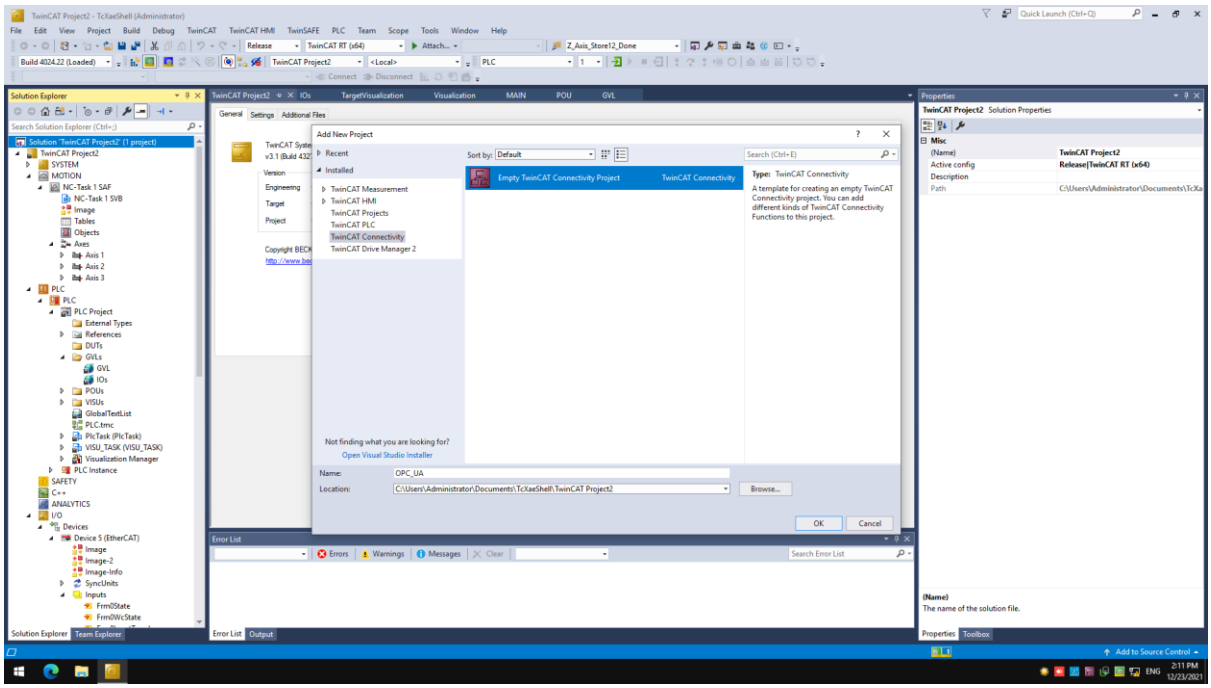

Link the created variable in the GVL list with the IOs

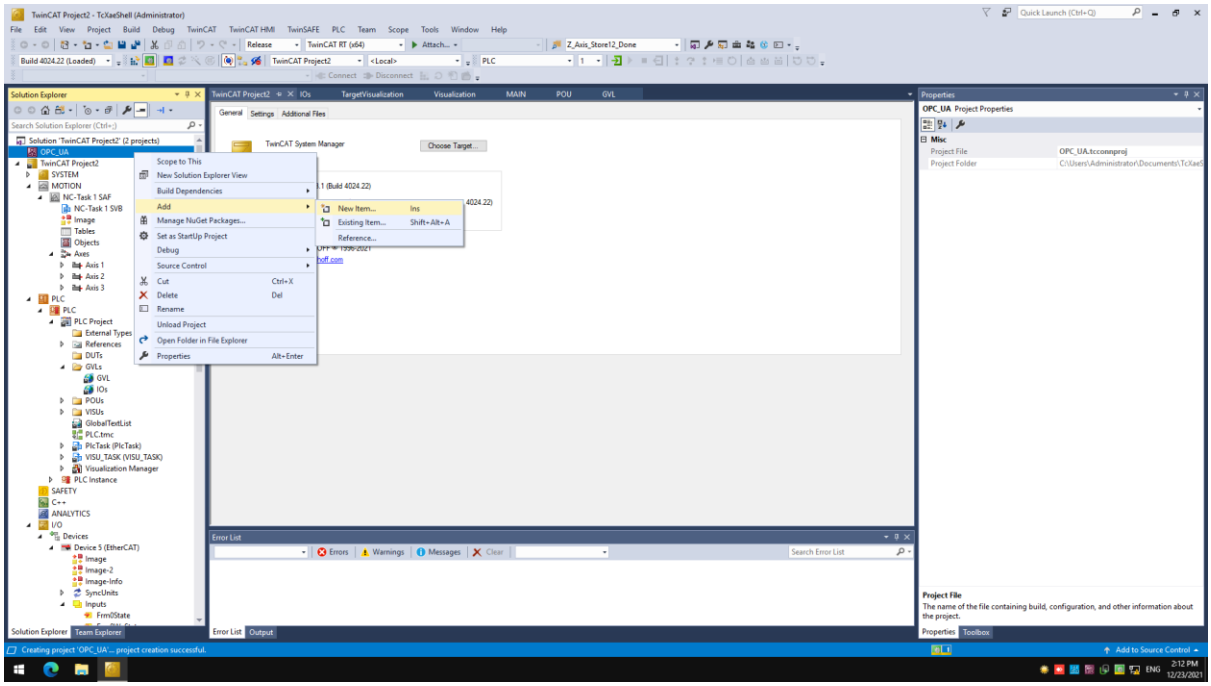

Once the variable is linked it will highlight in the box

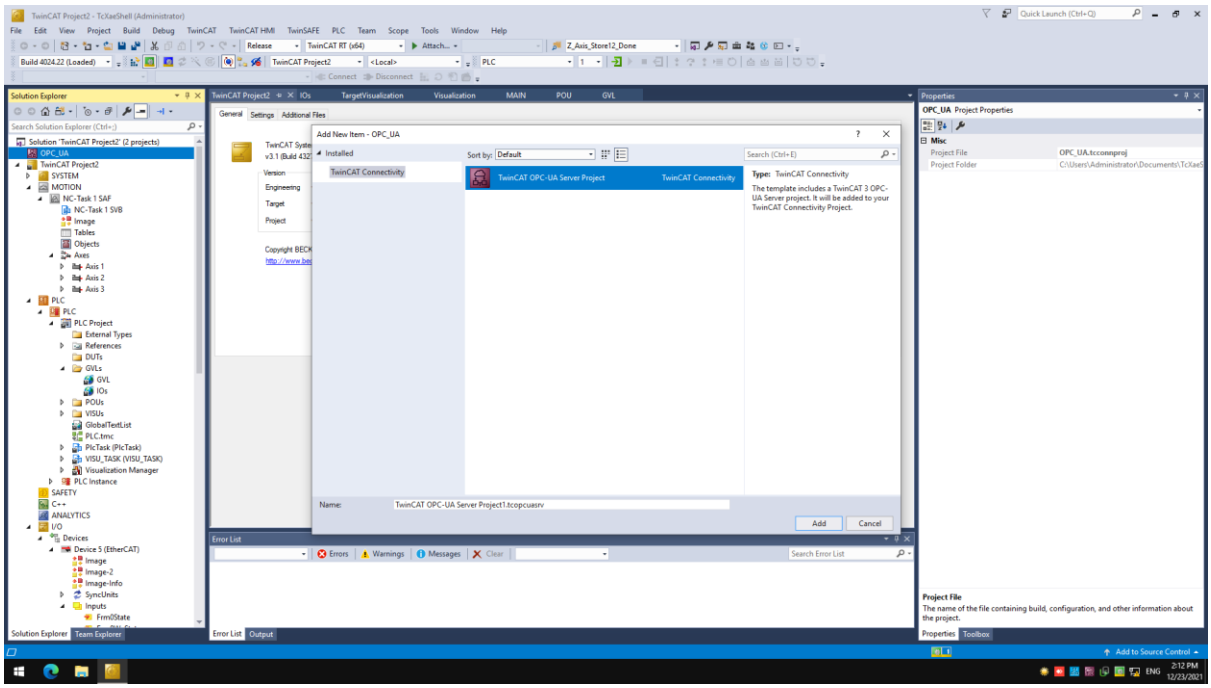
4.2 Creating OPC UA server and configuration: -

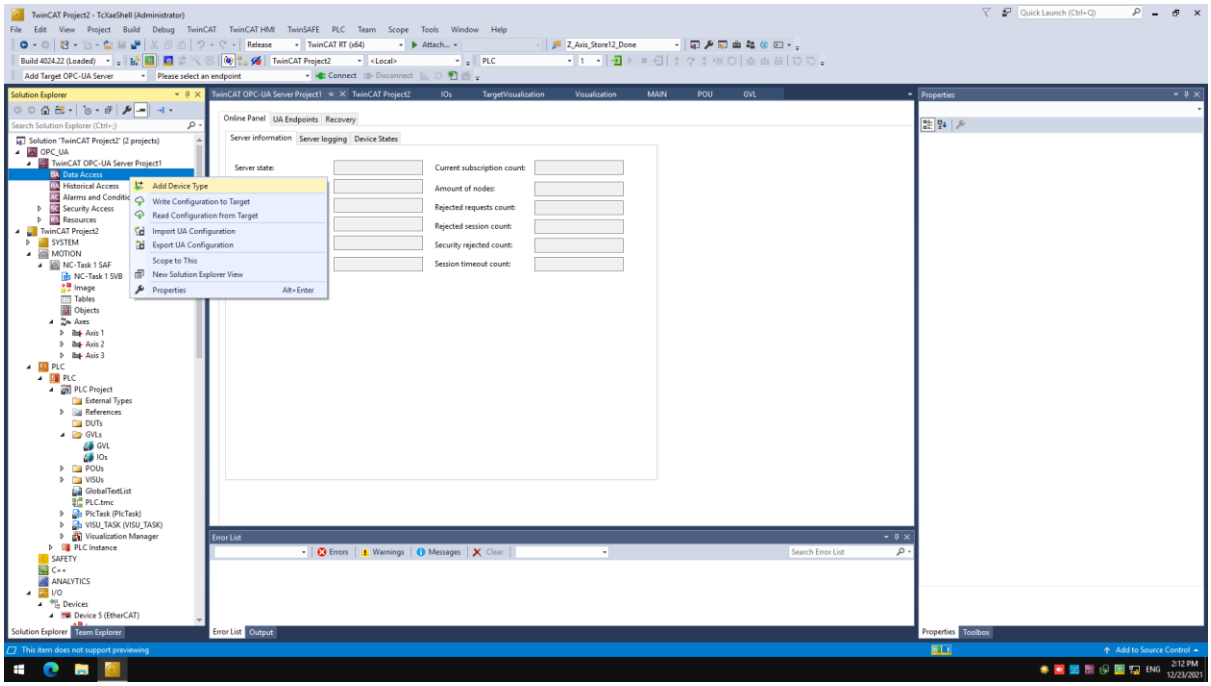

AMS Router
Local Computer
AMS Net Id:
OK
Cancel
BECKHOFF New Automa

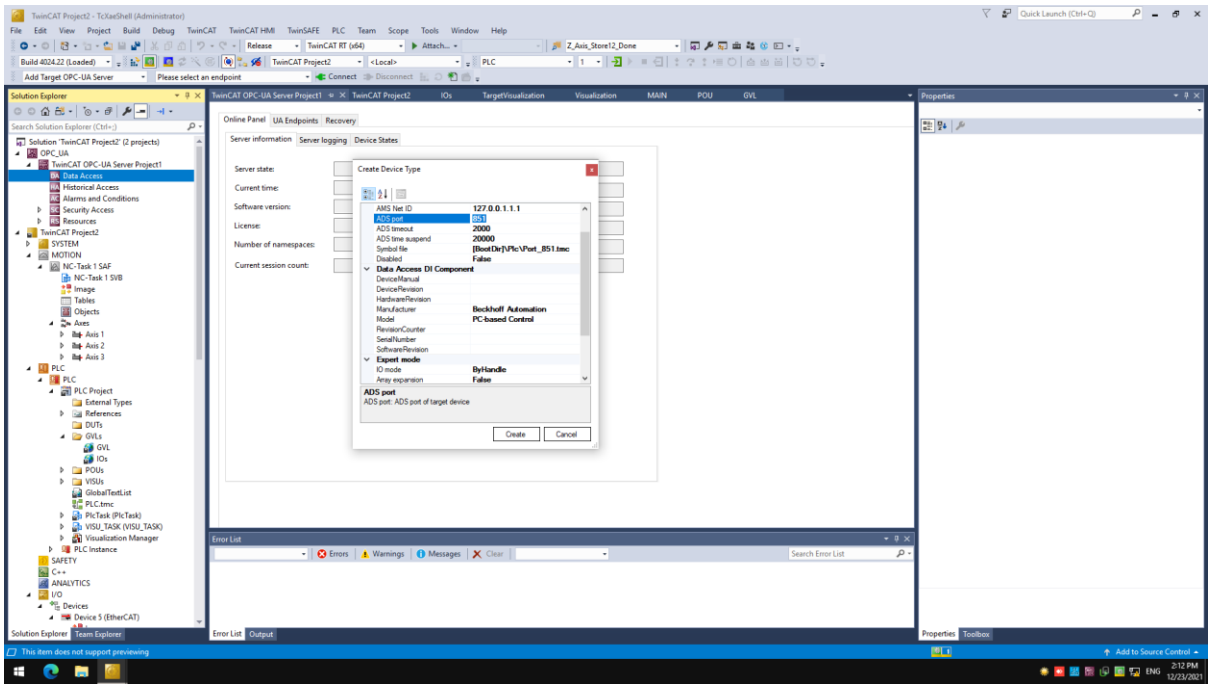

TwinCAT Project2 - TcXaeShell (Administrator)
Solution Explorer
Build Solution
Rebuild Solution
Clean Solution
Batch Build...
Configuration Manager...
Manage NuGet Packages for Solution...
New Solution Explorer View
Project Dependencies...
Project Build Order...
Add
Add Solution to Source Control...
Save TwinCAT Project2 as Archive...
Send TwinCAT Project2 by E-Mail...
Paste
Rename
Open Folder in File Explorer
Properties
New Project...
Existing Project...
New Item...
Existing Item...
New Solution Folder
Error List
Properties

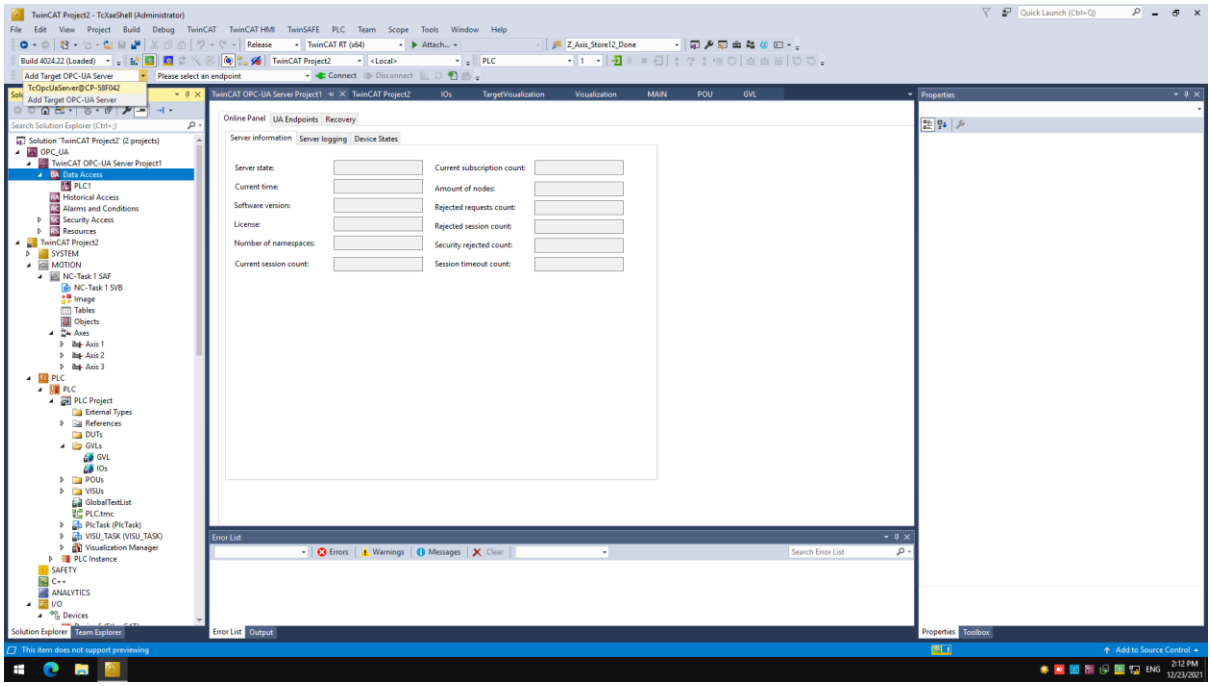

Add New Project
Empty TwinCAT Connectivity Project
OPC_UA
Solution Explorer
Properties

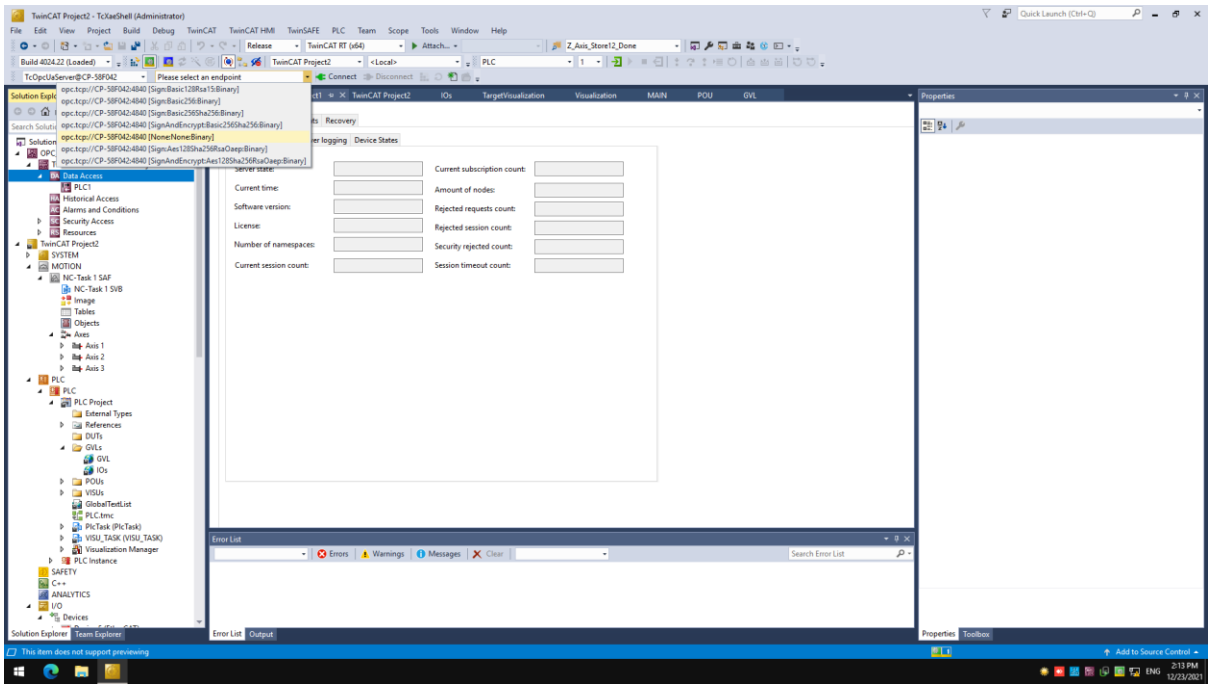

Add
New Item...
Existing Item...
Solution Explorer
Properties

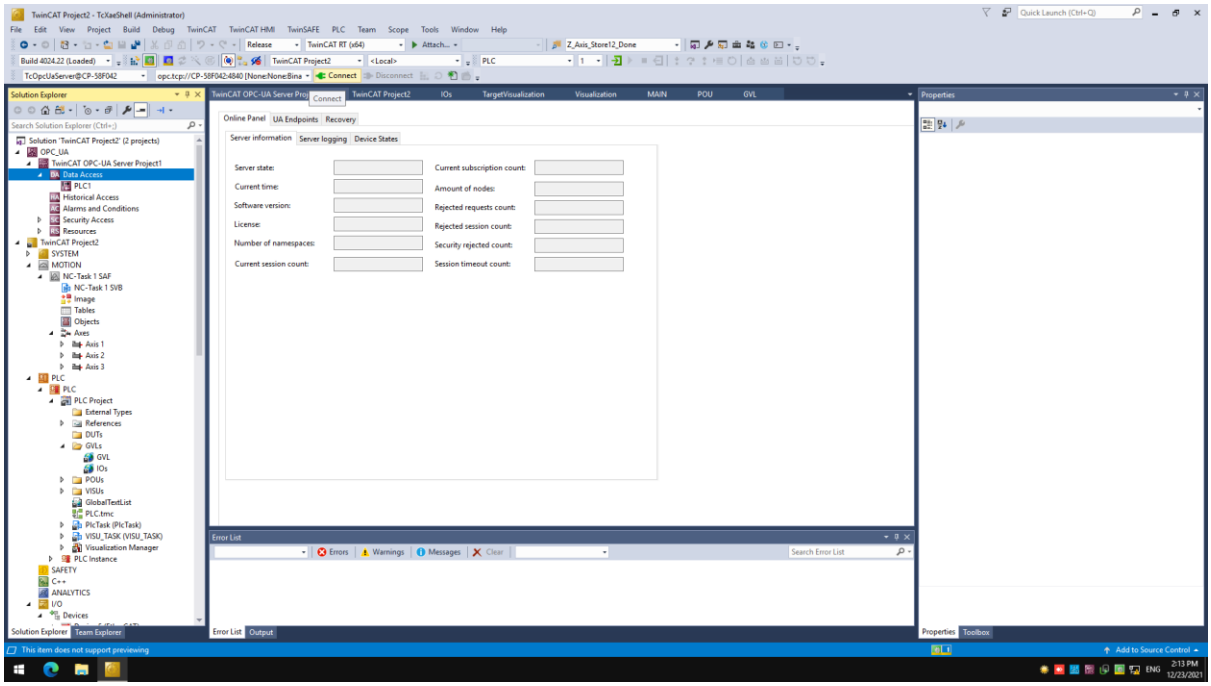

Add New Item - OPC_UA
TwinCAT OPC-UA Server Project
TwinCAT Connectivity
Add
Cancel

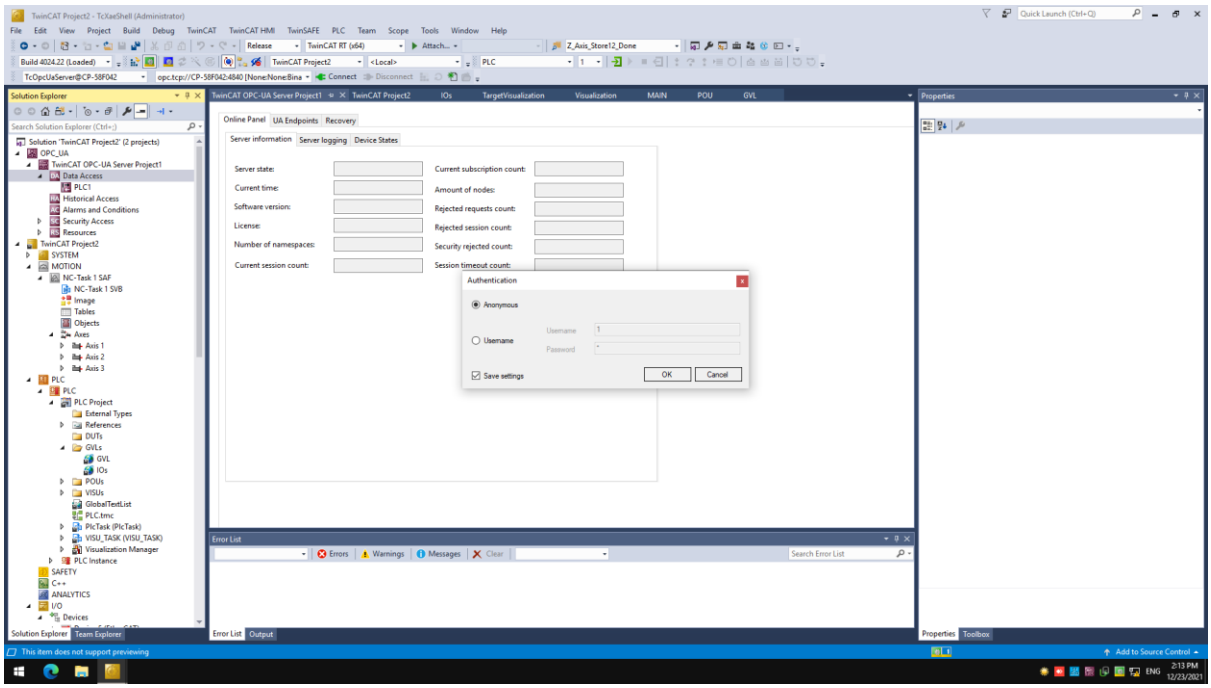

Add Device Type
Write Configuration to Target
Read Configuration from Target
Import UA Configuration
Export UA Configuration
Scope to This
New Solution Explorer View
Properties Alt+Enter
Server state:
Current subscription count:
Amount of nodes:
Rejected requests count:
Rejected session count:
Security rejected count:
Session timeout count:

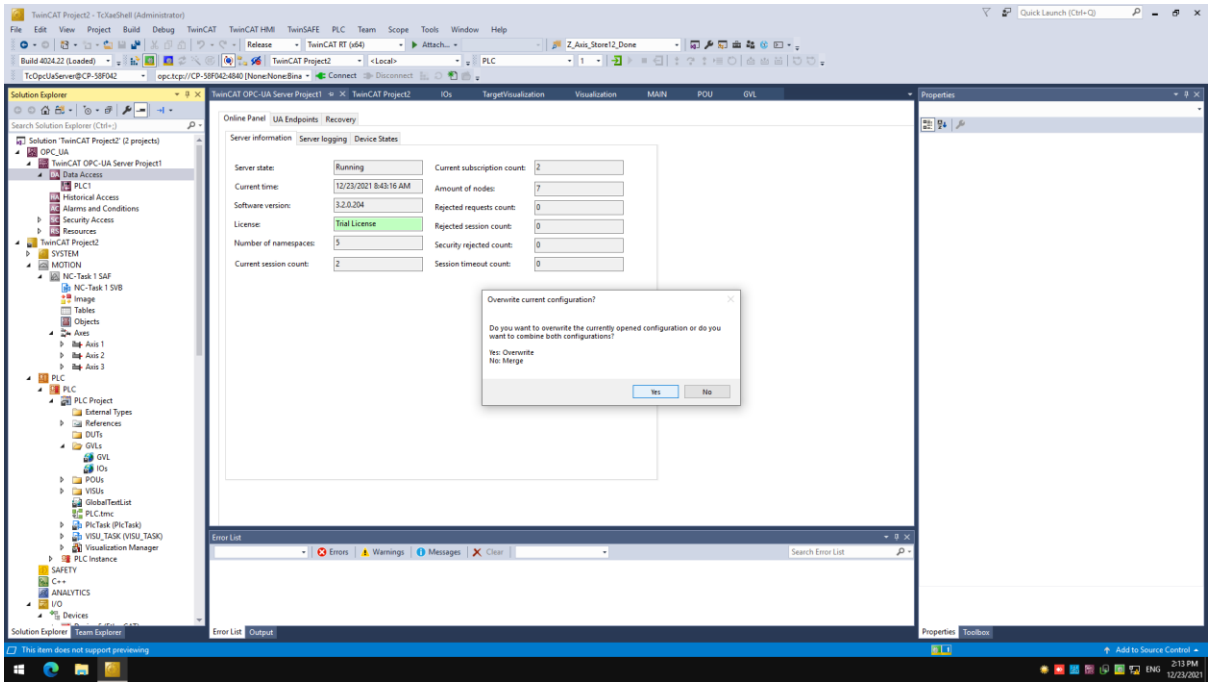

Create Device Type
AMS Net ID
127.0.0.1.1.1
ADS port
851
ADS timeout
2000
ADS time suspend
20000
Symbol file
[BootDir]\Plc\Port_851.tmc
Disabled
False
Data Access DI Component
DeviceManual
DeviceRevision
HardwareRevision
Manufacturer
Beckhoff Automation
Model
PC-based Control
RevisionCounter
SerialNumber
SoftwareRevision
Expert mode
IO mode
ByHandle
Array expansion
False
ADS port
ADS port: ADS port of target device
Create
Cancel

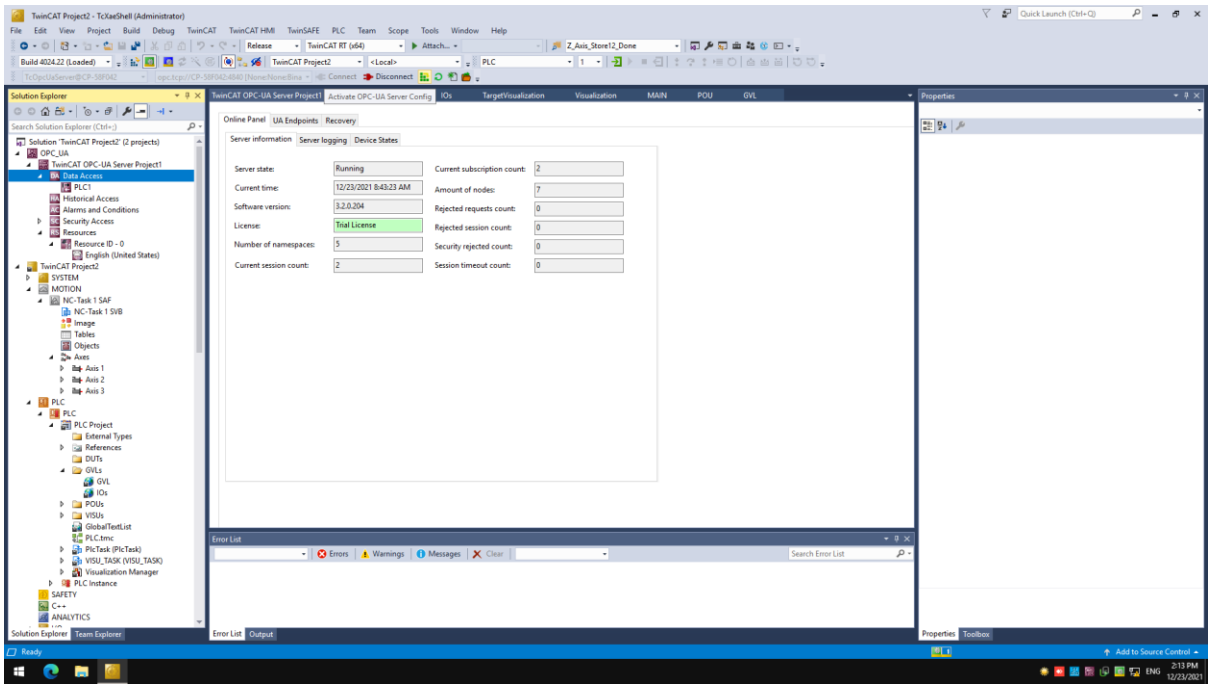


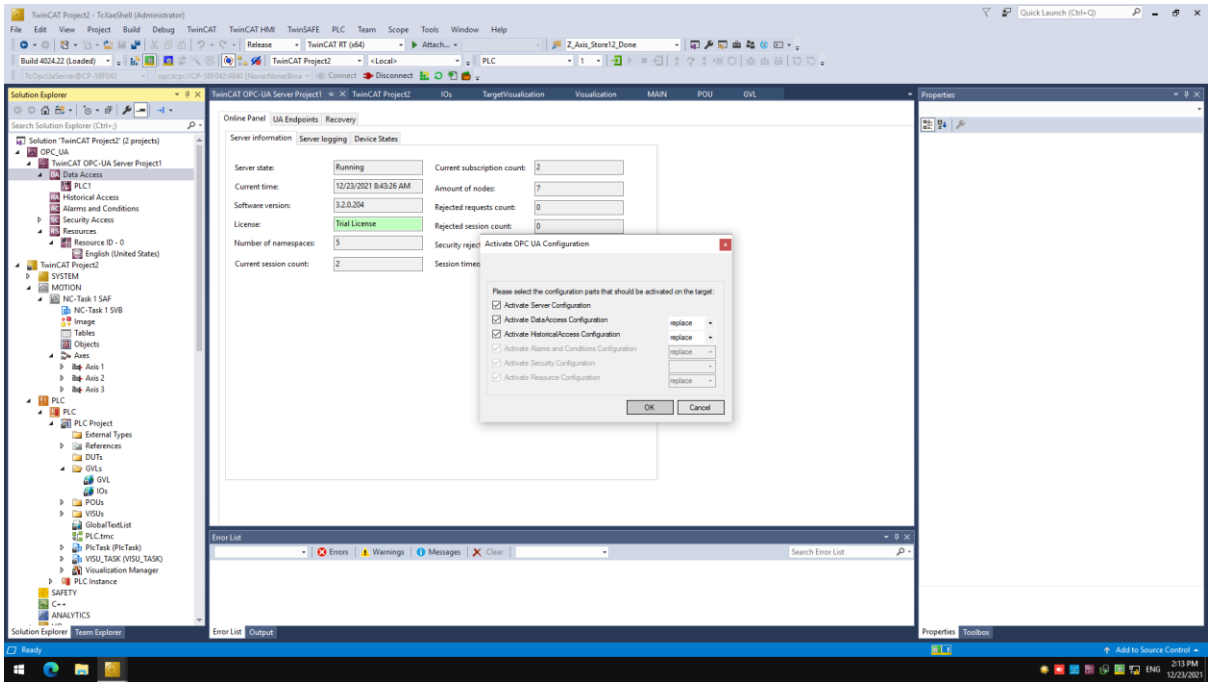

TwinCAT Project2 - TcXaeShell (Administrator)
Please select an endpoint
opc.tcp://CP-58F042:4840 [None:None:Binary]
Server state:
Current time:
Software version:
License:
Number of namespaces:
Current session count:
Current subscription count:
Amount of nodes:
Rejected requests count:
Rejected session count:
Security rejected count:
Session timeout count:

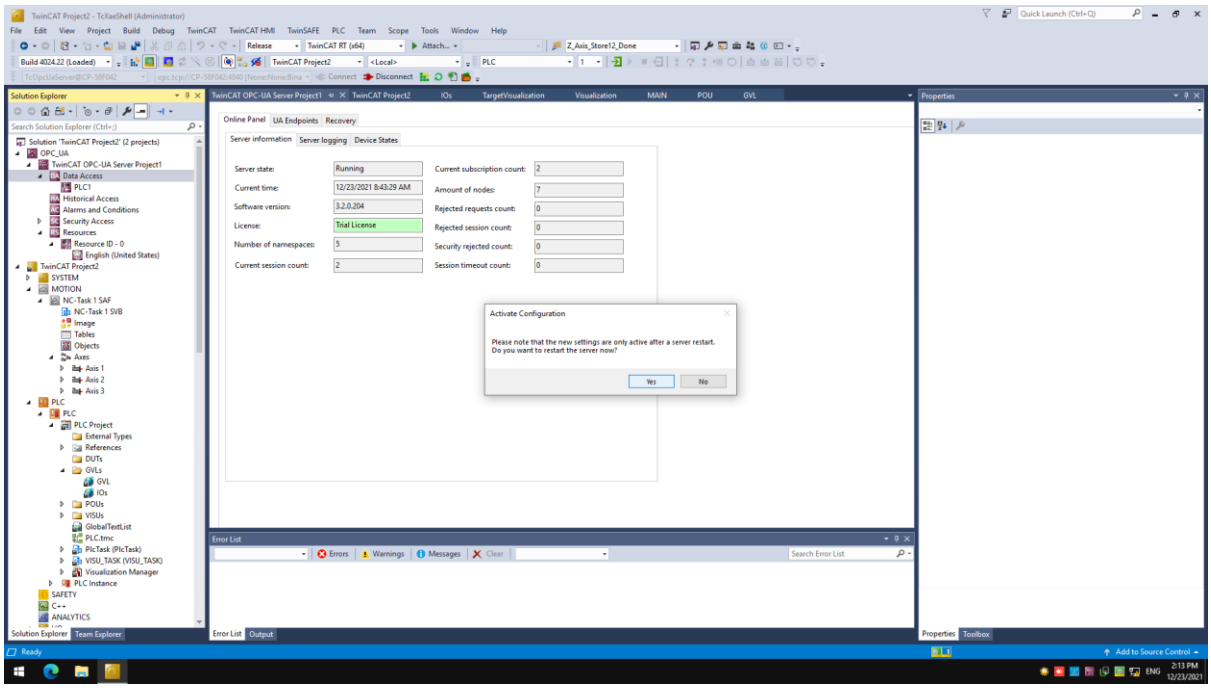

TwinCAT Project2 - TcXaeShell (Administrator)
Connect
Online Panel
UA Endpoints
Recovery
Server information
Server logging
Device States
Server state:
Current time:
Software version:
License:
Number of namespaces:
Current session count:
Current subscription count:
Amount of nodes:
Rejected requests count:
Rejected session count:
Security rejected count:
Session timeout count:

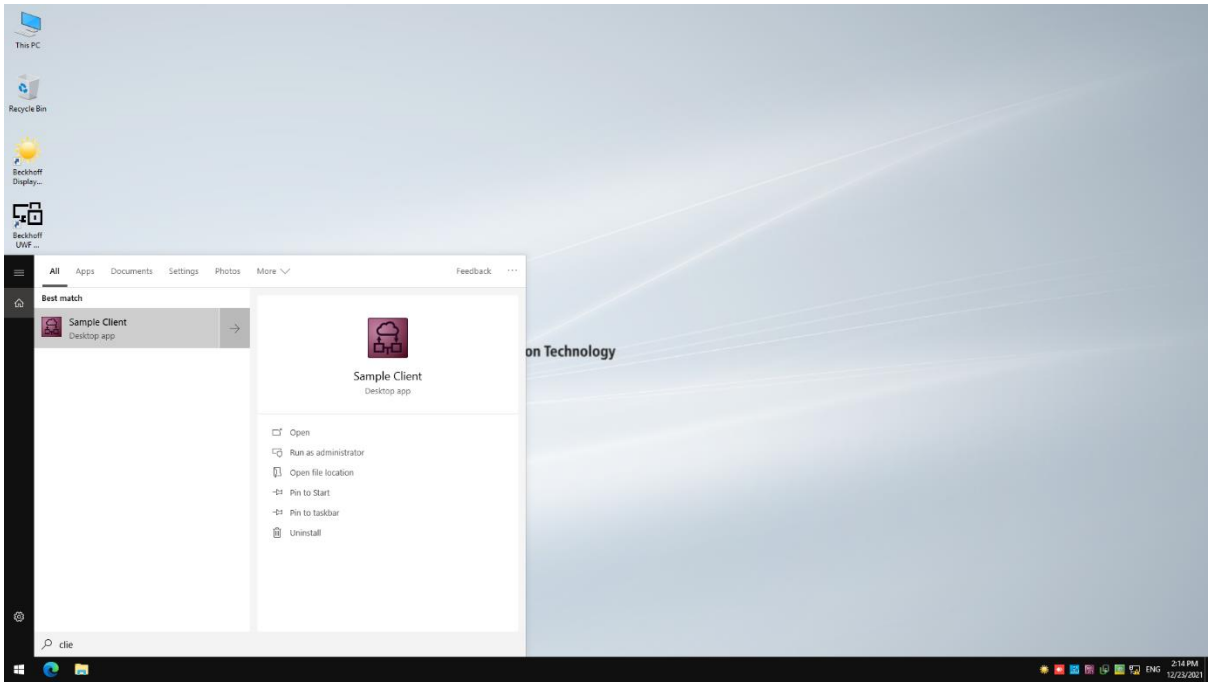

Authentication
Anonymous
Username
Save settings
OK
Cancel

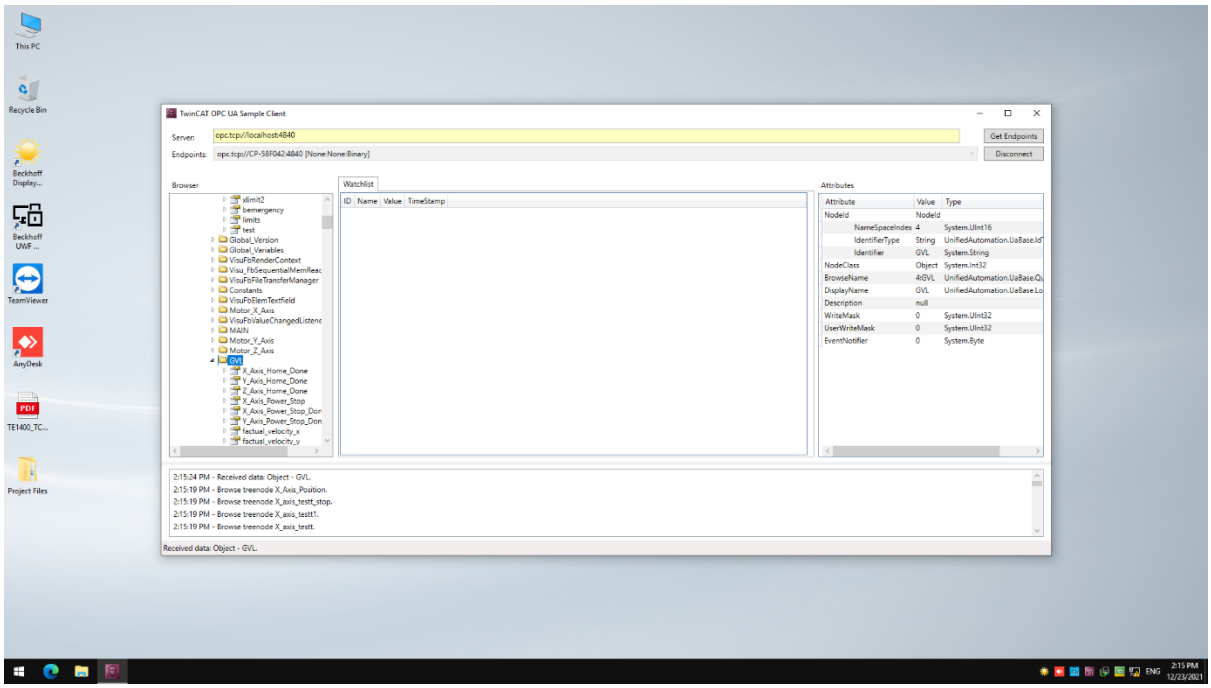

Overwrite current configuration?
Do you want to overwrite the currently opened configuration or do you want to combine both configurations?
Yes: Overwrite
No: Merge
Yes
No

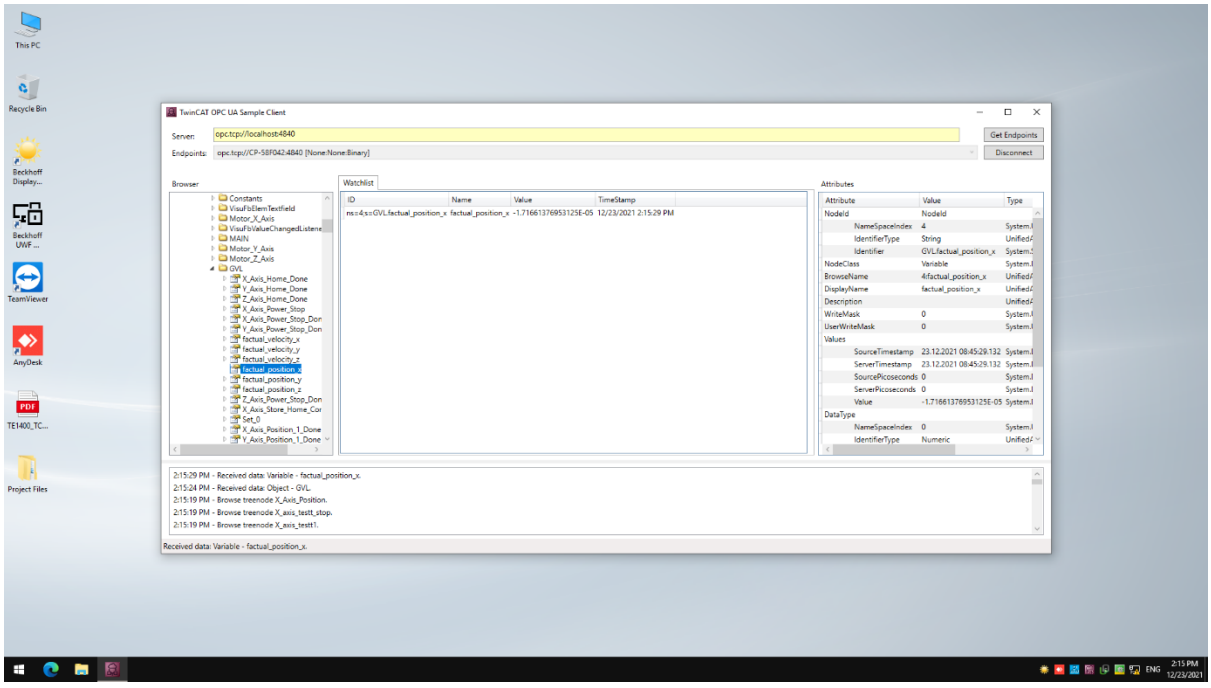





TwinCAT Project2 - TcXaeShell (Administrator)
Solution Explorer
TwinCAT OPC-UA Server Project1
Online Panel
UA Endpoints
Recovery
Server information
Server logging
Device States
Server state:
Running
Current time:
12/23/2021 8:43:29 AM
Software version:
3.2.0.204
License:
Trial License
Number of namespaces:
Current session count:
Current subscription count:
Amount of nodes:
Rejected requests count:
Rejected session count:
Security rejected count:
Session timeout count:
Activate Configuration
Please note that the new settings are only active after a server restart.
Do you want to restart the server now?
Yes
No
Error List
Properties
Toolbox
Output
Ready

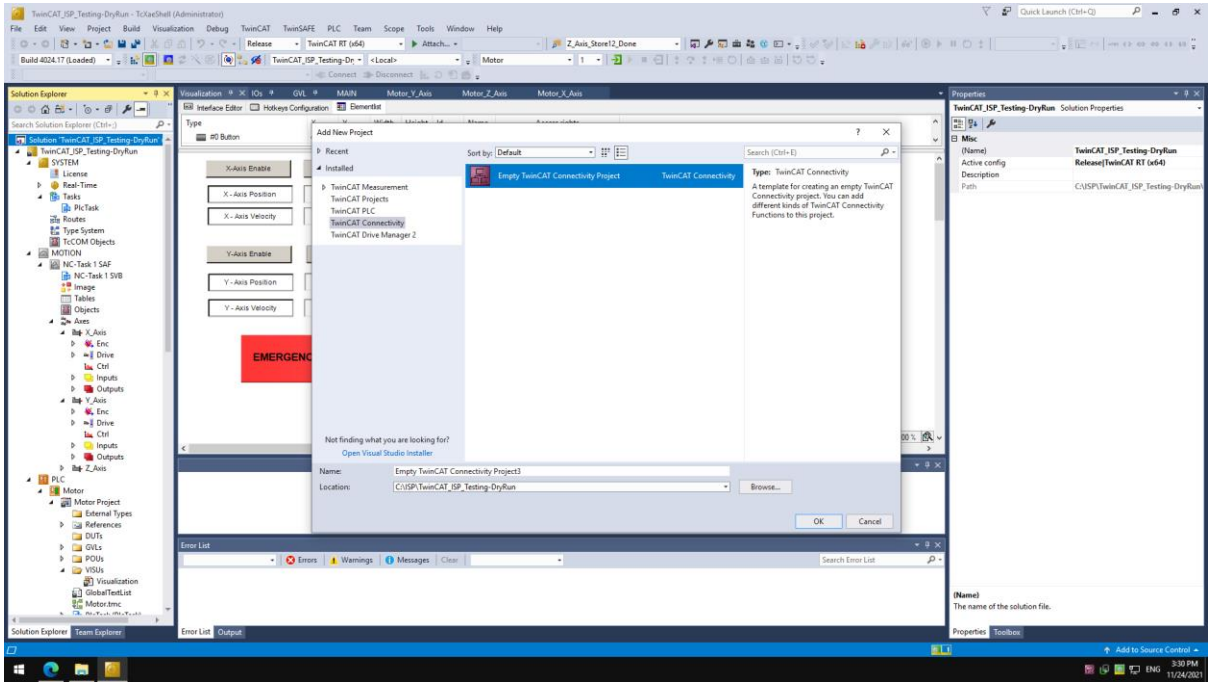

This PC
Recycle Bin
Beckhoff Display...
Beckhoff UWF...
All
Apps
Documents
Settings
Photos
More
Feedback
Best match
Sample Client
Desktop app
Open
Run as administrator
Open file location
Pin to Start
Pin to taskbar
Uninstall
clie

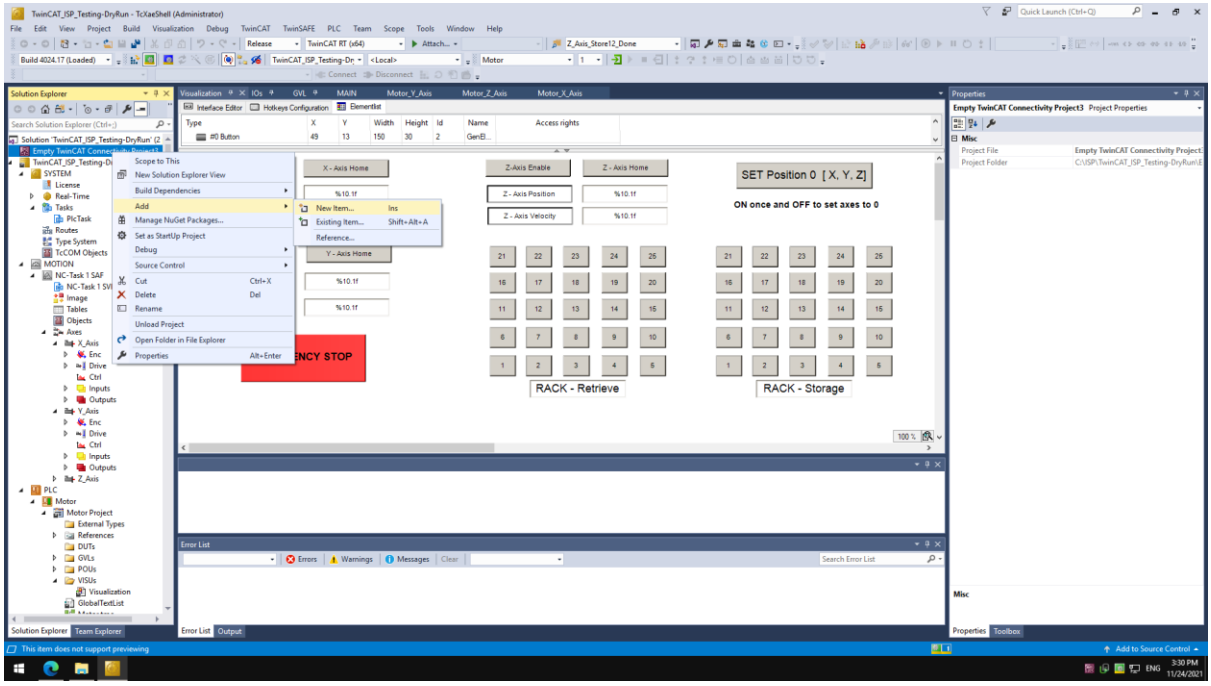

TwinCAT OPC UA Sample Client
Server:
opc.tcp://localhost:4840
Get Endpoints
Endpoints:
Disconnect
Browser
Watchlist
Attributes
2:15:24 PM - Received data: Object - GVL.
2:15:19 PM - Browse treenode X_Axis_Position.
2:15:19 PM - Browse treenode X_axis_testt_stop.
2:15:19 PM - Browse treenode X_axis_testt1.
2:15:19 PM - Browse treenode X_axis_testt.
Received data: Object - GVL.

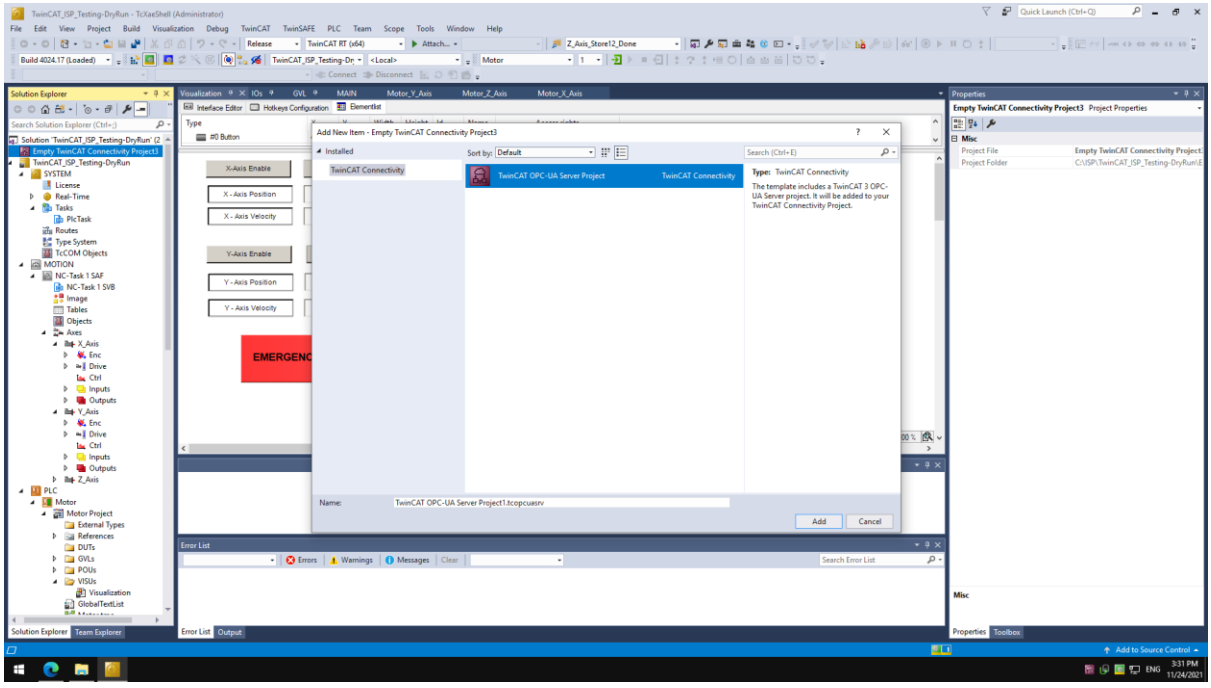

TwinCAT OPC UA Sample Client
Server:
opc.tcp://localhost:4840
Get Endpoints
Endpoints:
Disconnect
Browser
Watchlist
Attributes
2:15:29 PM - Received data: Variable - factual_position_x.
2:15:24 PM - Received data: Object - GVL.
2:15:19 PM - Browse treenode X_Axis_Position.
2:15:19 PM - Browse treenode X_axis_testt_stop.
2:15:19 PM - Browse treenode X_axis_testt1.
Received data: Variable - factual_position_x.

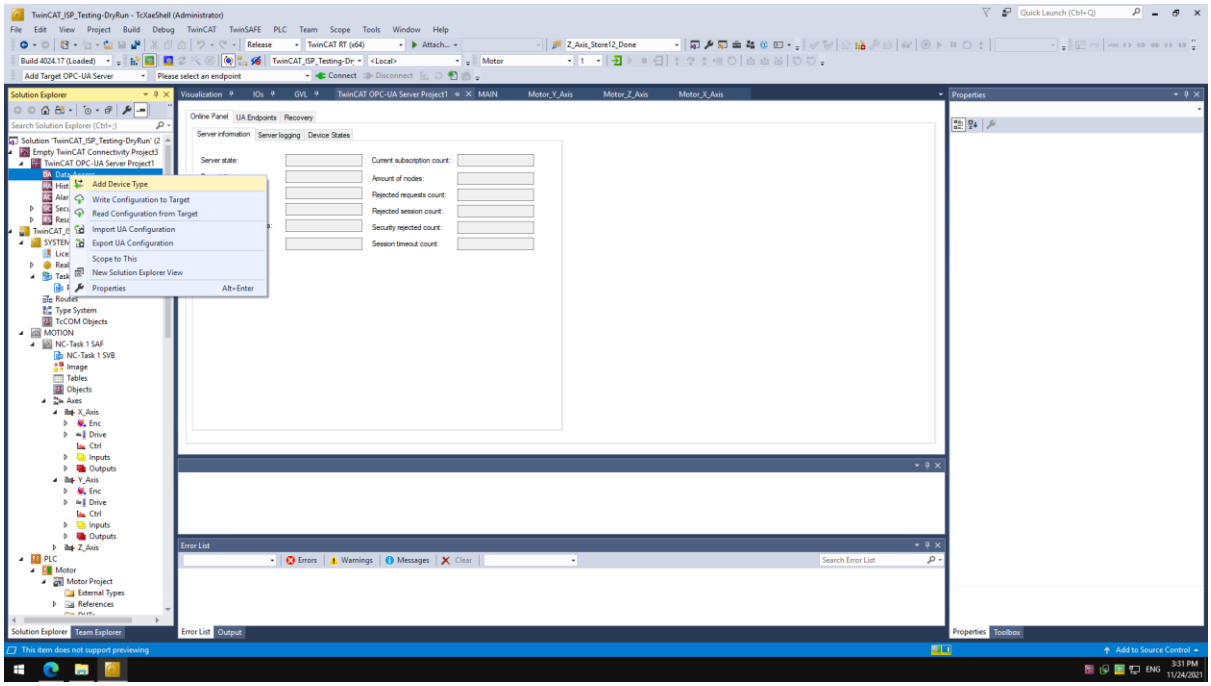
4.3 Digital Twin (MATLAB)

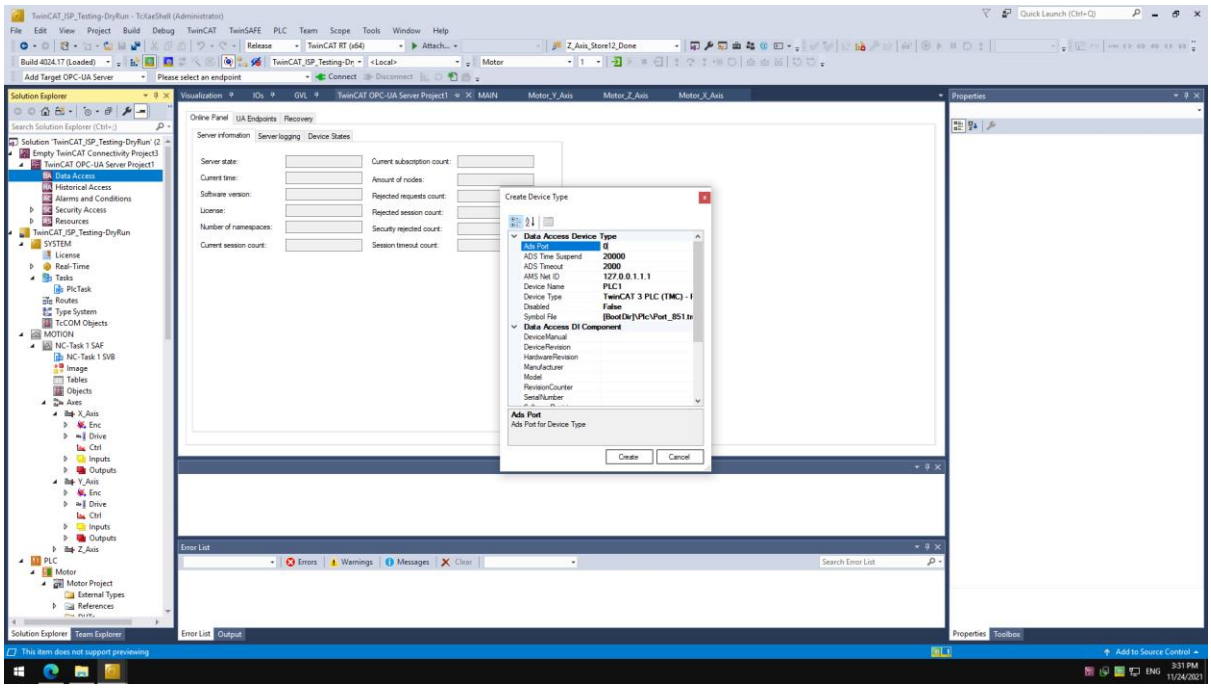
TwinCAT OPC Server setup: -

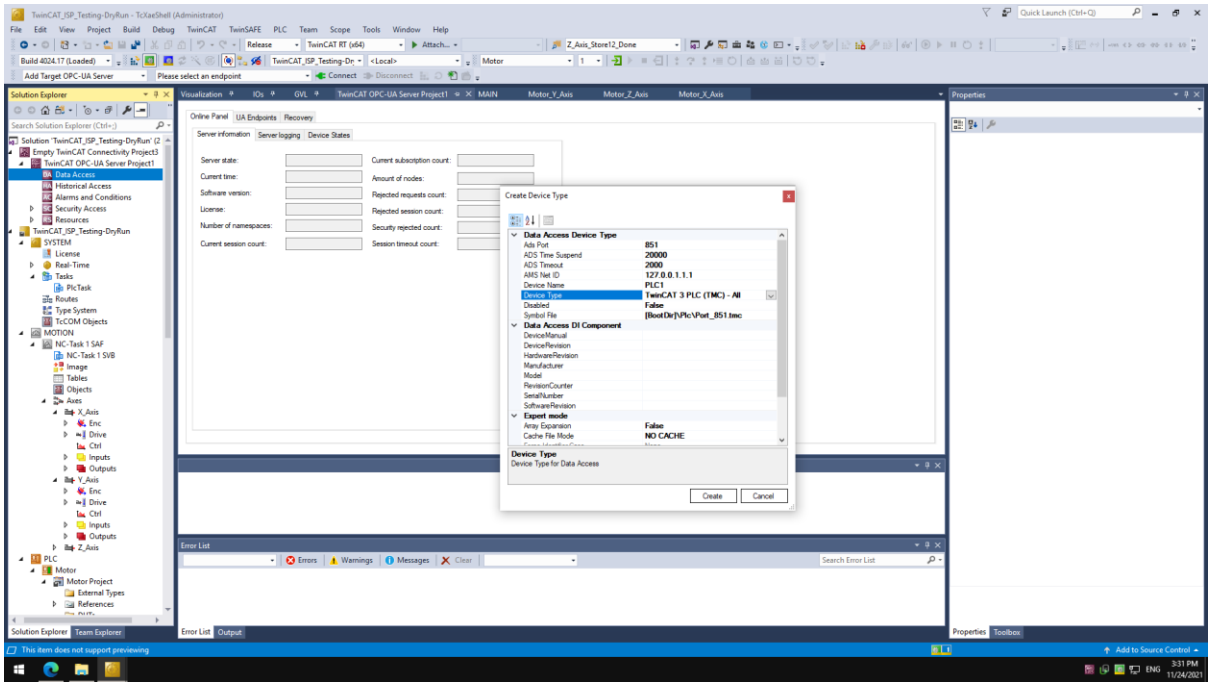


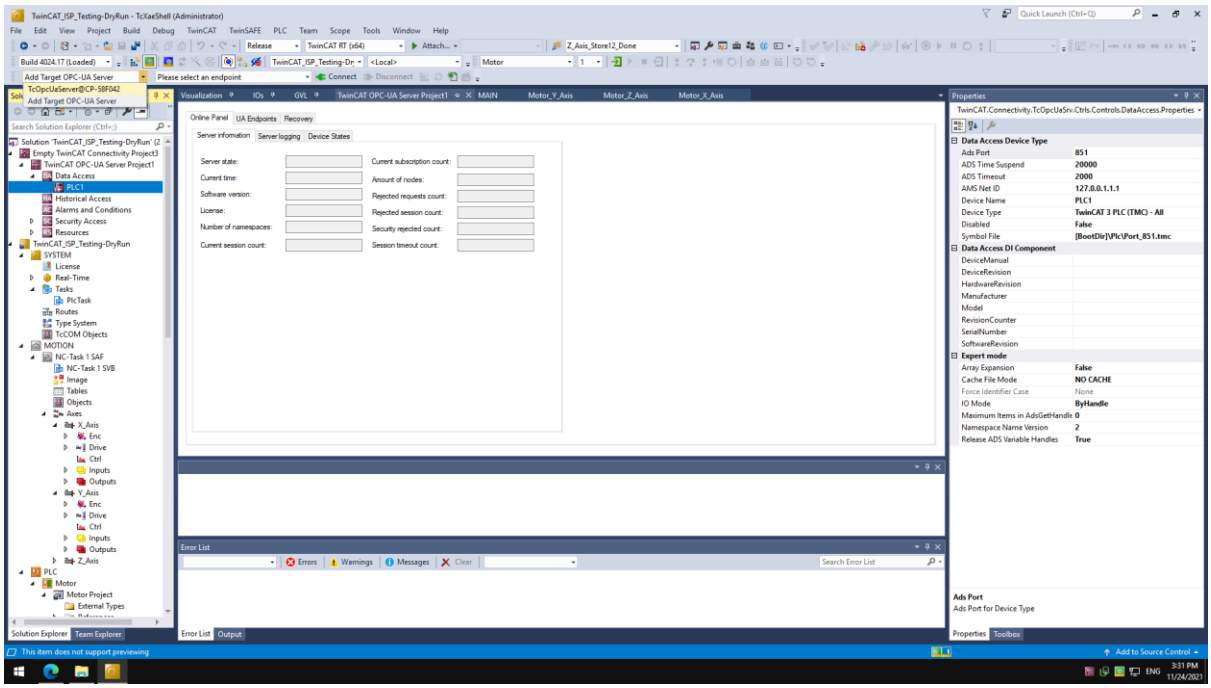


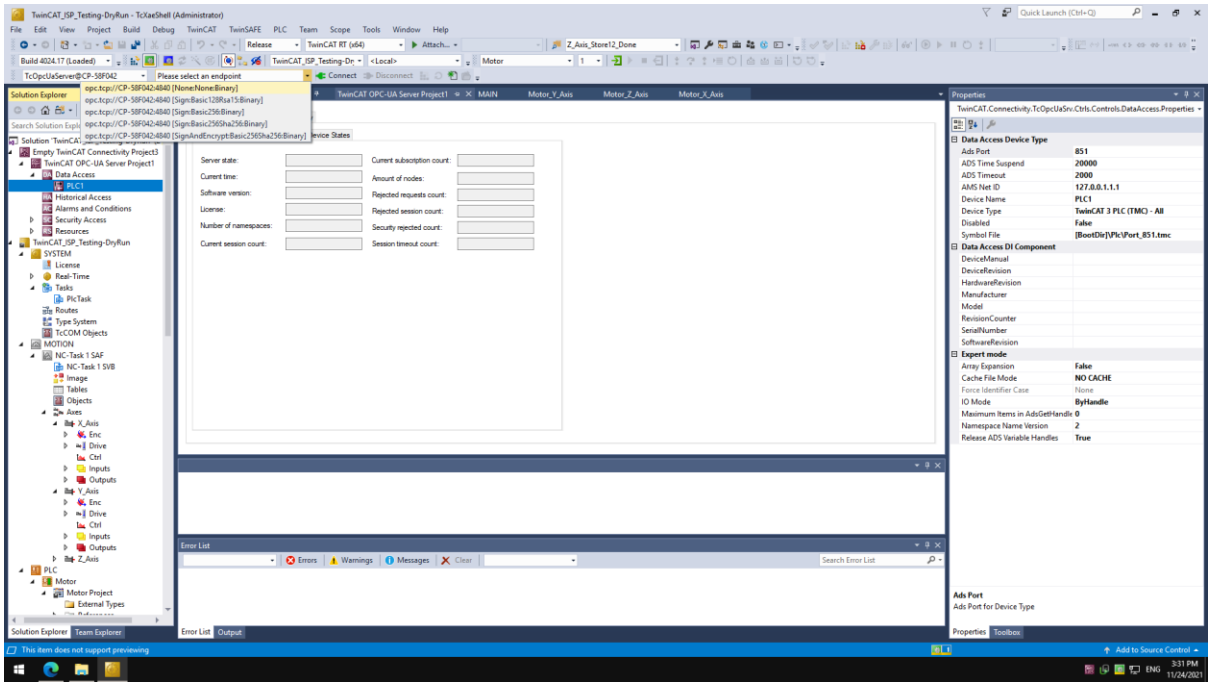

Add New Item - Empty TwinCAT Connectivity Project3
TwinCAT OPC-UA Server Project
TwinCAT Connectivity
Add
Cancel

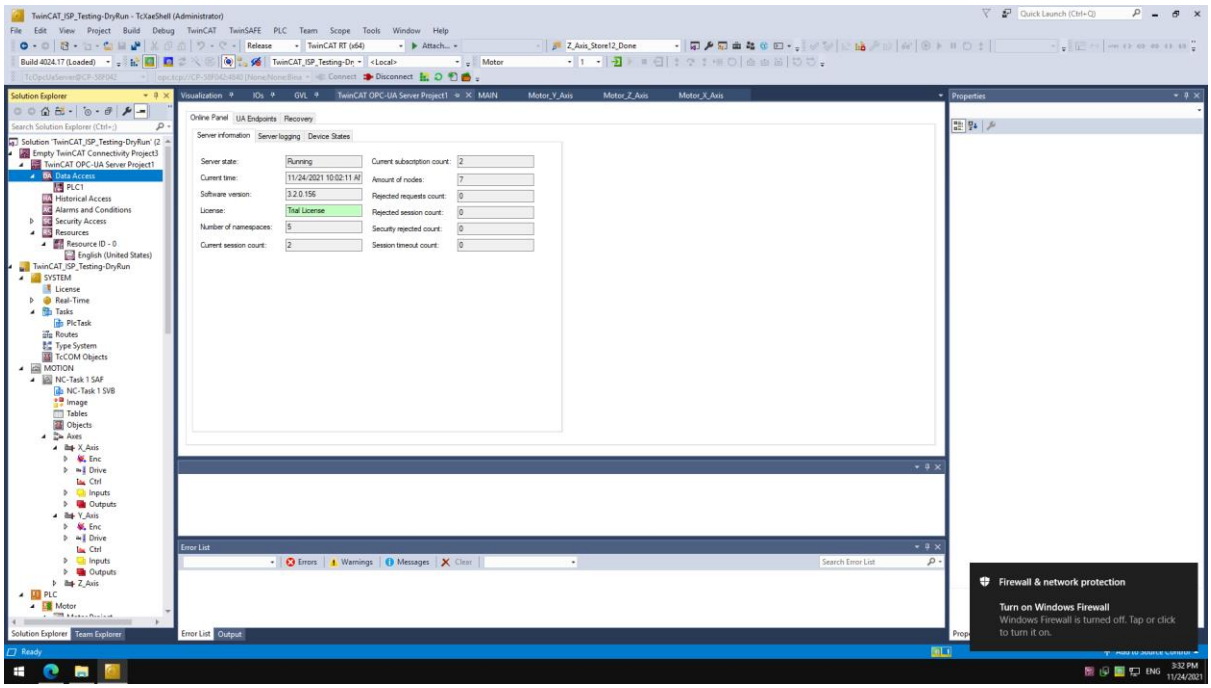

Add Device Type
Write Configuration to Target
Read Configuration from Target
Import UA Configuration
Export UA Configuration
Scope to This
New Solution Explorer View
Properties
Alt+Enter

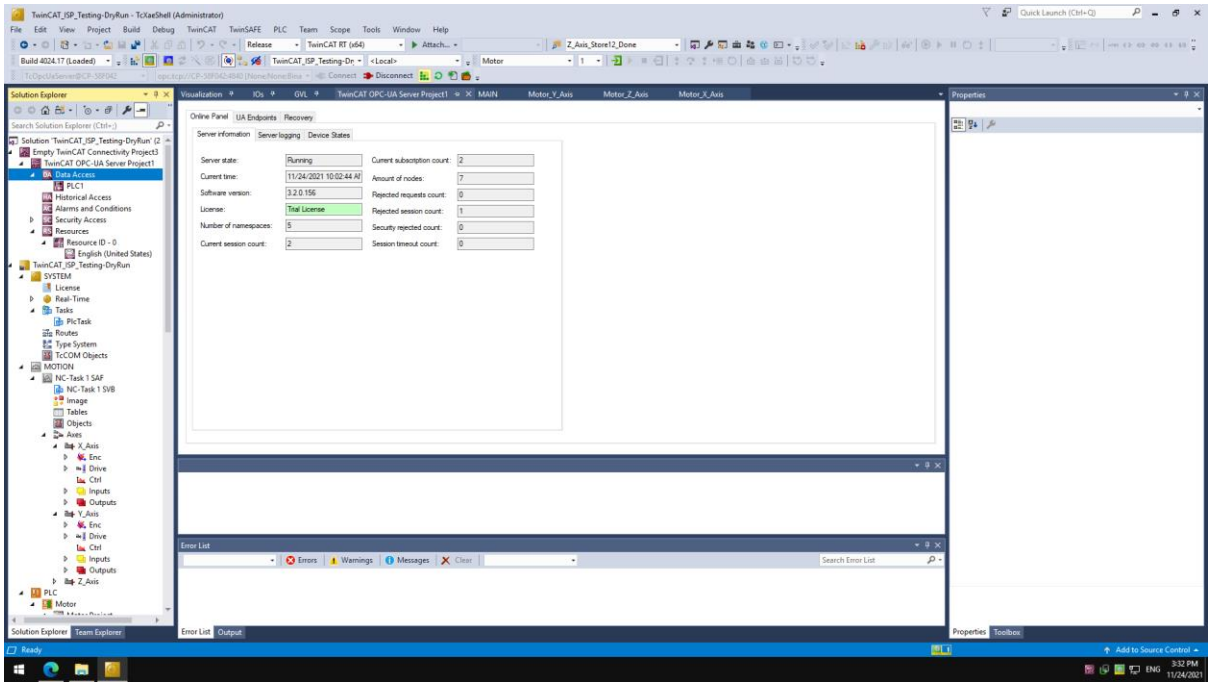

Create Device Type
Data Access Device Type
Ads Port
ADS Time Suspend
ADS Timeout
AMS Net ID
Device Name
Device Type
Disabled
Symbol File
Data Access DI Component
DeviceManual
DeviceRevision
HardwareRevision
Manufacturer
Model
RevisionCounter
SerialNumber
Ads Port
Ads Port for Device Type
Create
Cancel


Create Device Type
Data Access Device Type
Ads Port
ADS Time Suspend
ADS Timeout
AMS Net ID
Device Name
Device Type
Disabled
Symbol File
Data Access DI Component
DeviceManual
DeviceRevision
HardwareRevision
Manufacturer
Model
RevisionCounter
SerialNumber
SoftwareRevision
Expert mode
Array Expansion
Cache File Mode
Device Type
Device Type for Data Access
Create
Cancel

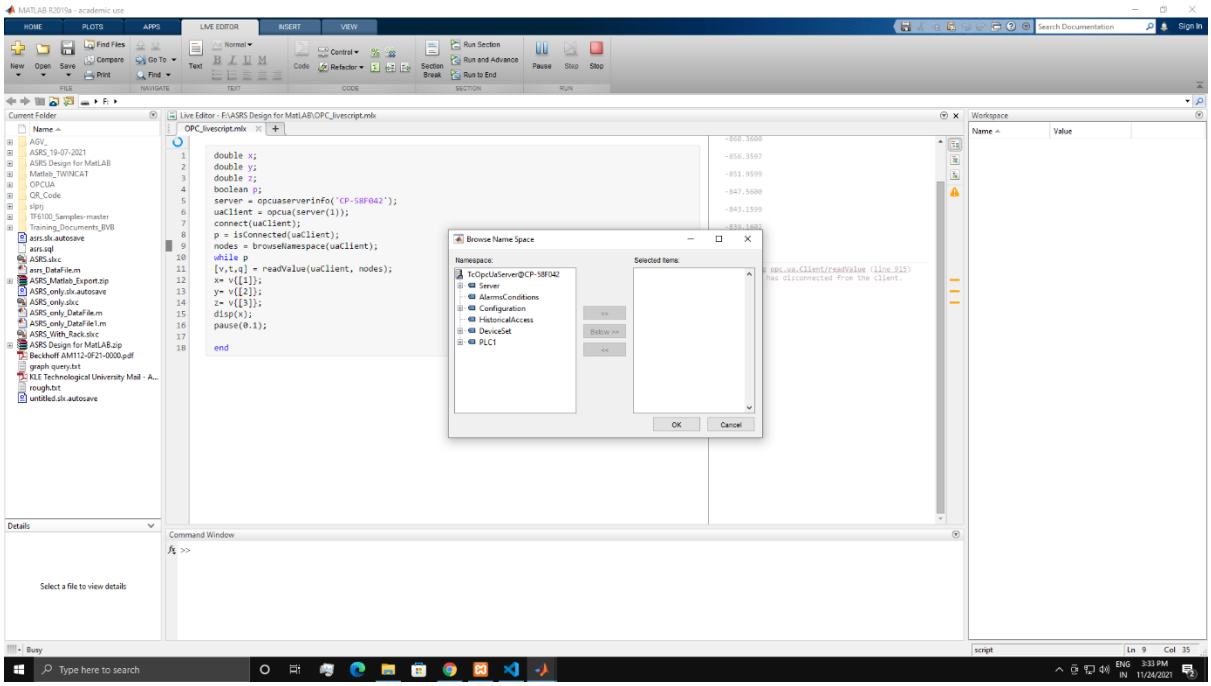


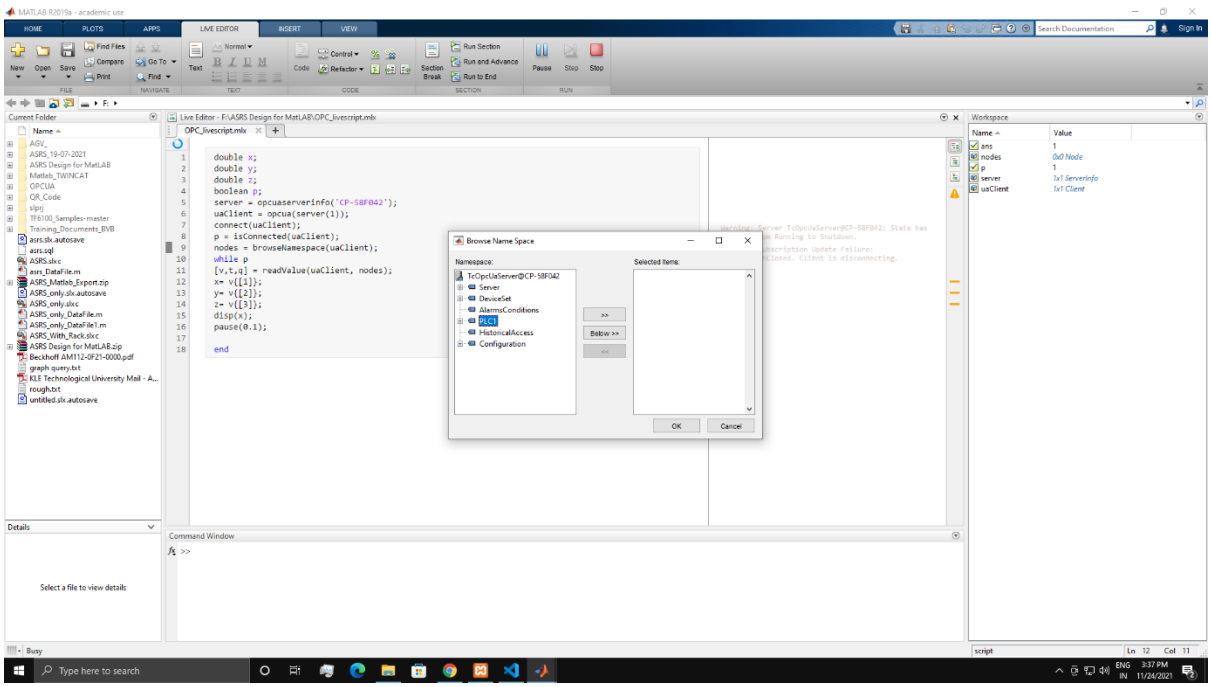


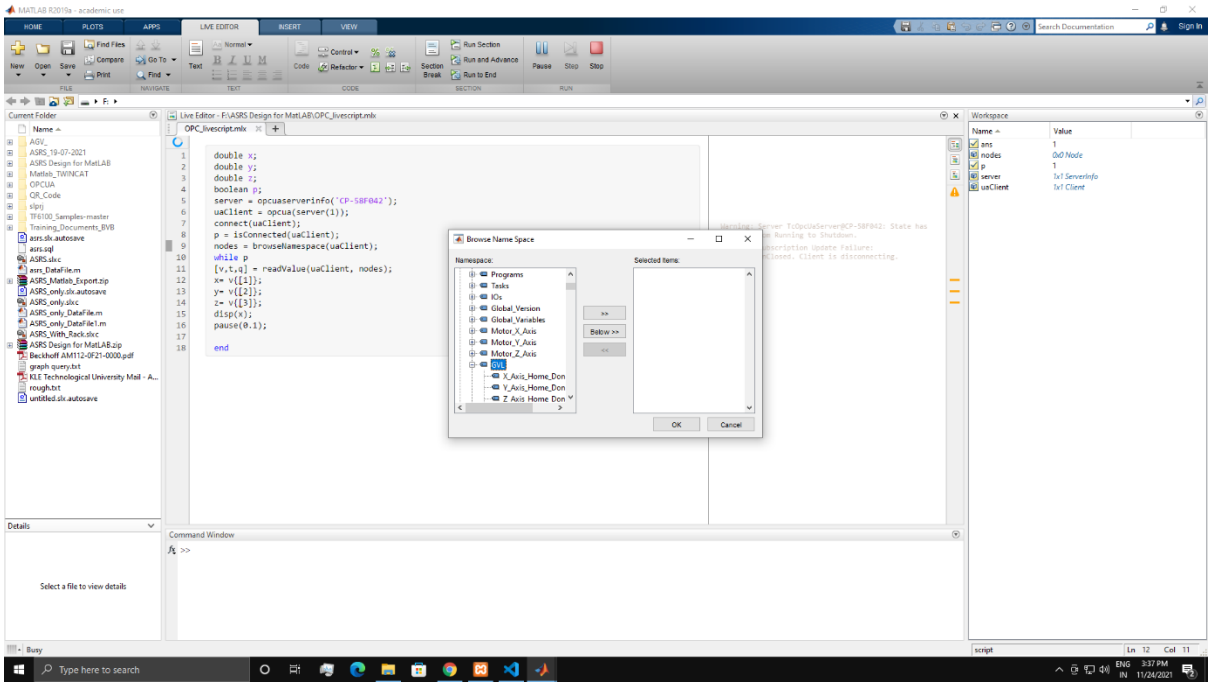

TwinCAT_ISP_Testing-DryRun - TcXaeShell (Administrator)
Firewall & network protection
Turn on Windows Firewall
Windows Firewall is turned off. Tap or click to turn it on.

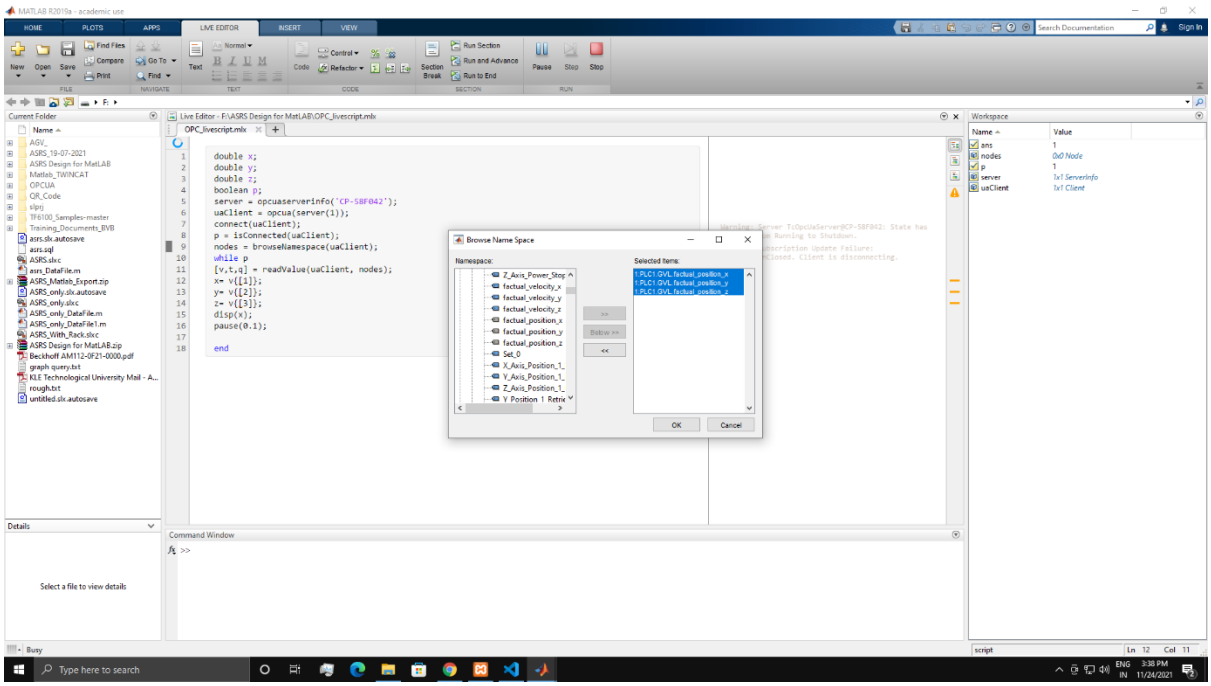

TwinCAT_ISP_Testing-DryRun - TcXaeShell (Administrator)

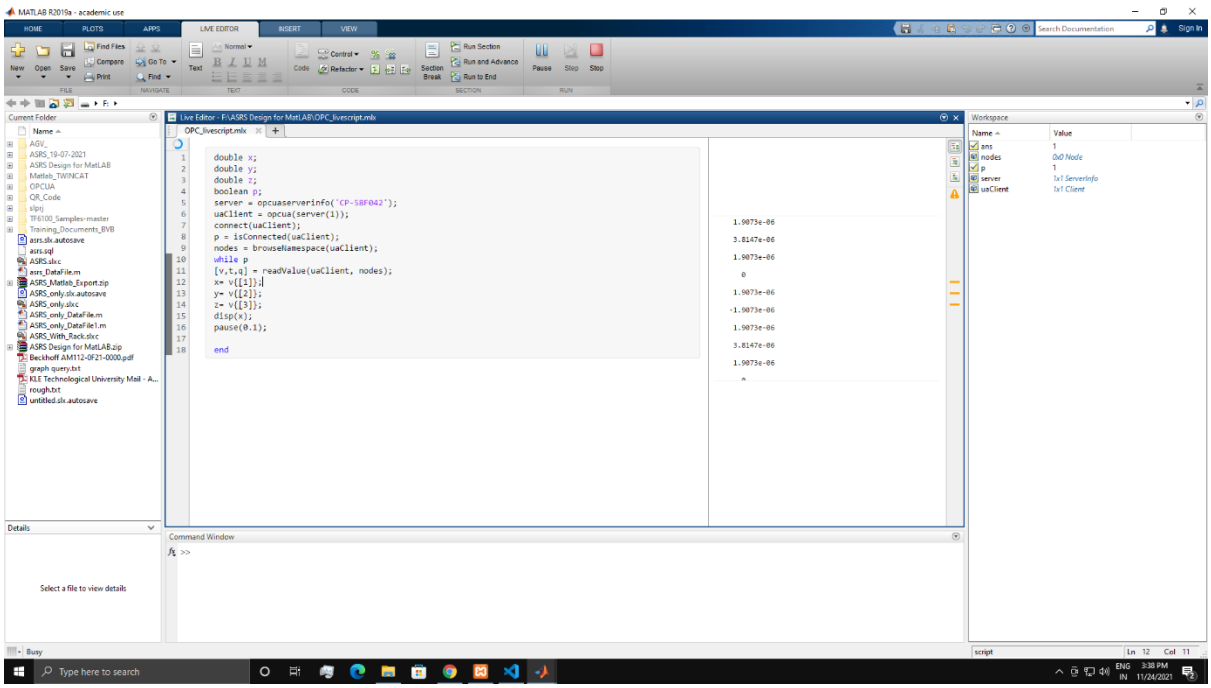
4.3.1 MATLAB setup: -

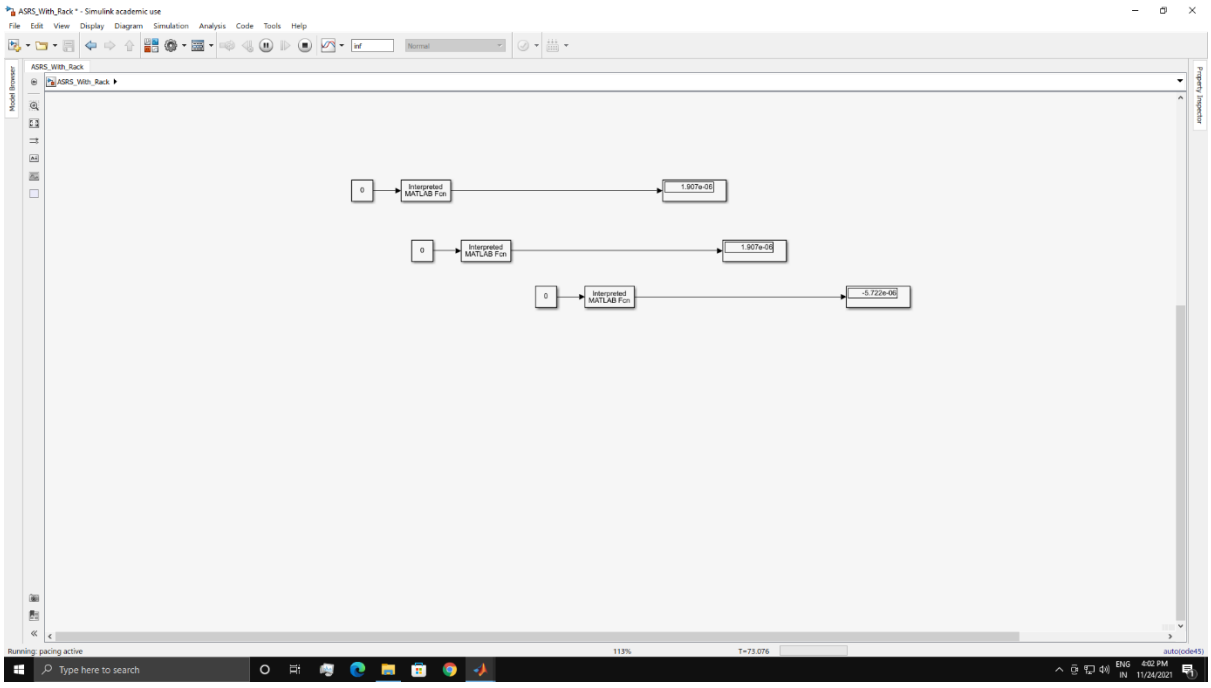


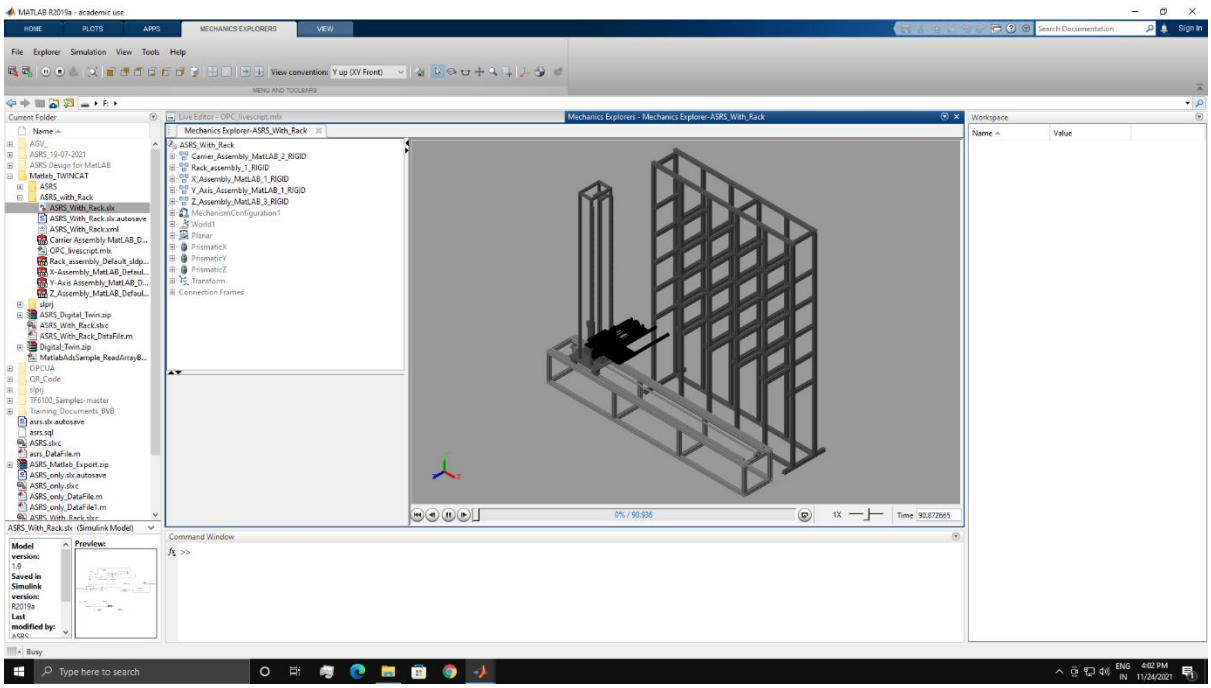


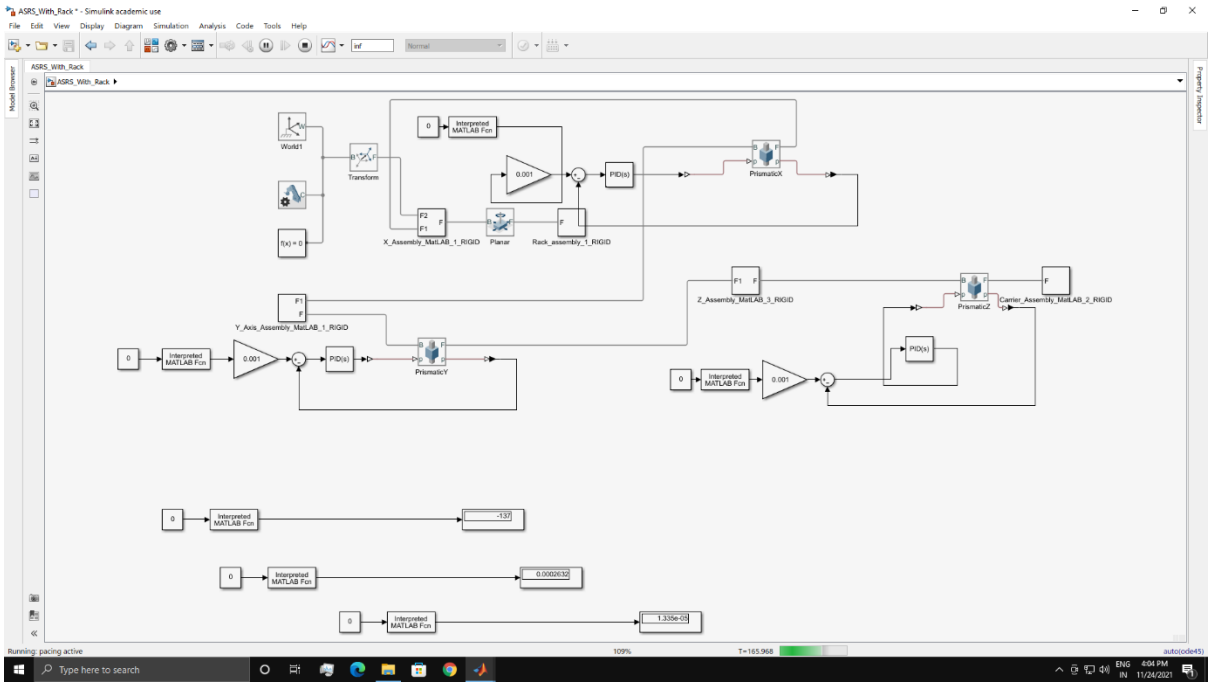

Browse Name Space
Namespace:
Selected Items:
OK
Cancel


Browse Name Space
Namespace:
Selected Items:
OK
Cancel


MATLAB R2019a - academic use
Live Editor - F:\ASRS Design for MatLAB\OPC_livescript.mlx
OPC_livescript.mlx
double x;
double y;
double z;
boolean p;
server = opcuaserverinfo('CP-58F042');
uaClient = opcua(server(1));
connect(uaClient);
p = isConnected(uaClient);
nodes = browseNamespace(uaClient);
while p
[v,t,q] = readValue(uaClient, nodes);
x= v([1]);
y= v([2]);
z= v([3]);
disp(x);
pause(0.1);
end
Workspace
Command Window
Select a file to view details


ASRS_With_Rack * - Simulink academic use
ASRS_With_Rack
Interpreted MATLAB Fcn
1.907e-06
Interpreted MATLAB Fcn
1.907e-06
Interpreted MATLAB Fcn
-5.722e-06

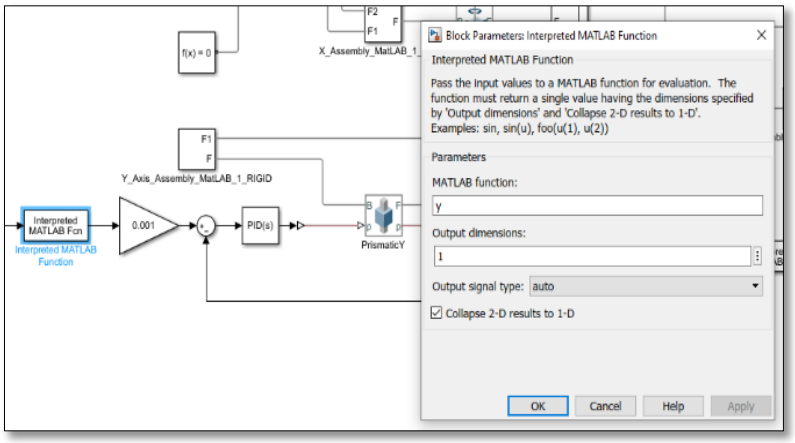




After the Solidworks model is exported to MATLAB using the Solidworks-Matlab extension, the .slx file is opened in Simulink. This file consists of the Simulink block diagram which is the code or function codes as blocks connected to different blocks, which combined give us a working model.

Here, various blocks like constants, joints, gains, PID controllers, solid files, and etcetera are present which cover various domains like arithmetic, logical, trigonometry, mechanics to kinematics, etc.

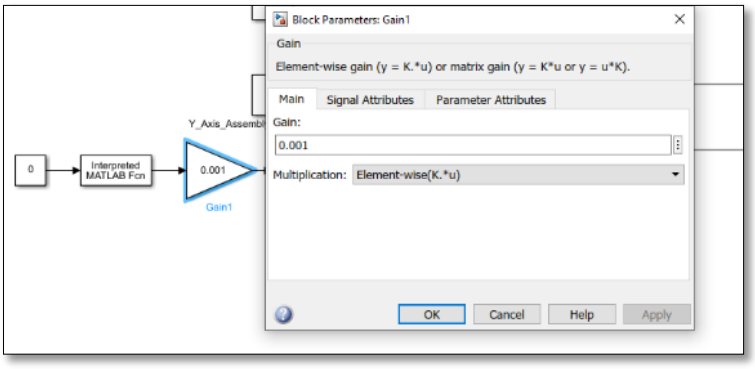
Below listed are the required blocks, their definitions and their use to achieve a fully functional Digital Twin:



Integrated MATLAB function:

This block is used in order to access a MATLAB workspace variable into Simulink environment. Here, the integrated MATLAB function is used to access 'y' variable which is generated in the workspace after receiving the variables from the OPC server.

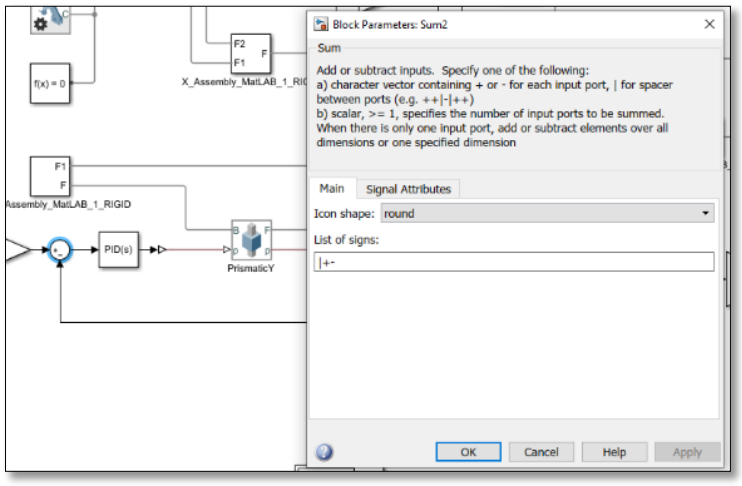
This received variable are the factual position of x, y and z axes. Variables with the same name are created in the workspace and then are accessed into the Simulink model for the Digital twin. This variable has integer values which show the position in mms.



Gain:

The gain block is used as a multiplier to the input. It basically multiplies the input with the given gain and generates output.

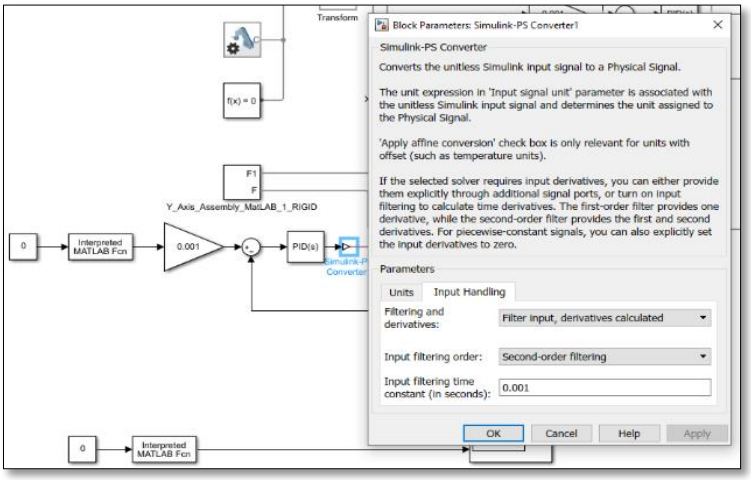
Here, 0.001 is used for gain as the input is factual position in mm which needs to be converted to mtrs as the joints need the position in mtrs.



Summing block:

Similar to the gain block, the summing block, takes two input values and adds them up and outputs the sum of those variables. When the block is expanded, the list of signs, consists a pipeline followed by the sign of the values.

Here, the signs are |+- because the feedback loop is always taken in with a negative sign while the forward loop is positive.

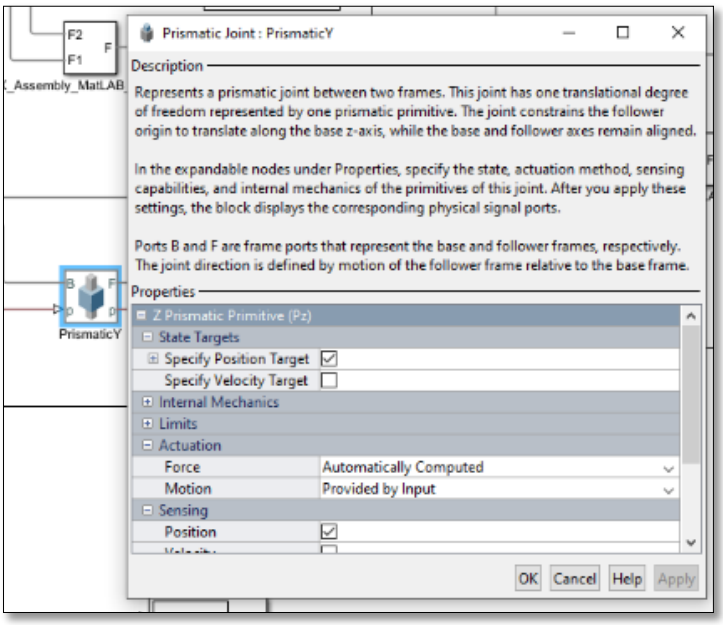


Simulink- PS (convertor):

The Simulink-PS convertor is used whenever the Simulink values are to be converted to Physical Signals to be accessed by the joints.

The filtering and derivatives is toggled to “Filter input, derivatives calculated” , the input filtering order to “Second order filtering” and Input filtering time constant to 0.001.

This will make the inputs the second order derivatives and will be sampled every 0.001 seconds giving more precise digital twin.



Prismatic joint:

The prismatic joint is used whenever translation is to be performed. The motion takes place along the Z axis.

When the block is expanded, there are various options which are to be changed for proper adaption for the digital twin. The state targets is the input which is being taken. Here, the specify position target is checked, as the input being given is the factual position in mms.

Here, the motion is being provided by input from us as we have the factual position from the OPC server while the force is automatically computed.

Sensing is the outputs being taken out from the joint block. Here , only the position is required as the output, as it is feeding the value back to the summing block for the feedback loop.

Settings required before simulating the Digital Twin:

- The simulation time should be changed to “inf” i.e. infinity from the definite time.
- The pacing for the simulation is better if turned on.
  - Menu bar → Simulation → Pacing →Check box.
- The python script to connect MATLAB to the OPCUA server should be run after the Simulink simulation is run. The vice versa is not possible.
- Select “Y up (XY front)” in view convention in the simulation explorer page.
- Isometric view show cases most of the features of the system. (better if selected)
- Always add all the folders and subfolders to the path before trying to run the .slx file. Else the simulation crashes or gives errors.

4.4 User Interface

A web based user interface has been developed to control and manipulate the ASRS from any remote location with user friendly interface which also provides adding into or getting information from the database.

4.4.1. Web Based

Various markup and programming languages used are:

HTML: Used for structuring and presenting content on the World Wide Web.

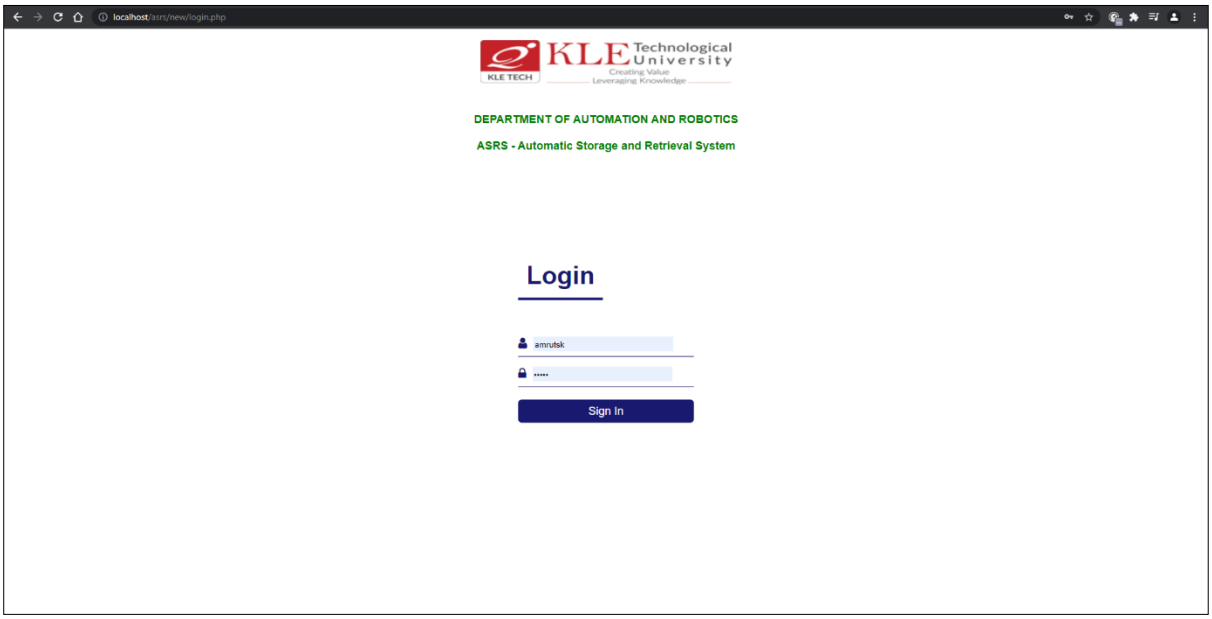
JavaScript: a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled and multiparadigm. It has dynamic typing, prototype-based object-orientation and first-class functions.

CSS: style sheet language used for describing the presentation of a document written in a mark-up language such as HTML.

PHP: general-purpose scripting language geared towards web development.

Ajax: a set of web development techniques that uses various web technologies on the client-side to create asynchronous web applications.

Bootstrap: CSS framework directed at responsive, mobile-first front-end web development.



Login page:

The login page is for allowance of only the registered users into the interface.

The registered users, employees here, should enter their credentials in order to access the site, ASRS and the database itself.

The page uses all the above-mentioned languages and is responsive. If the user successfully logs in, he/she gets access to the whole site. If the user is unable to log in successfully, a pop-up appears suggesting to retry logging in.

Once the person logs out of the site, it redirects to the same page and has to login again. This keeps in mind the security of the system so that no other individual can misuse the site.

Home page:

After logging in, the user redirects to the home page, where some visual aid is provided, such as bar graphs and histogram, which show the hourly empty-occupied locations of the day and daily storage-retrieves respectively.

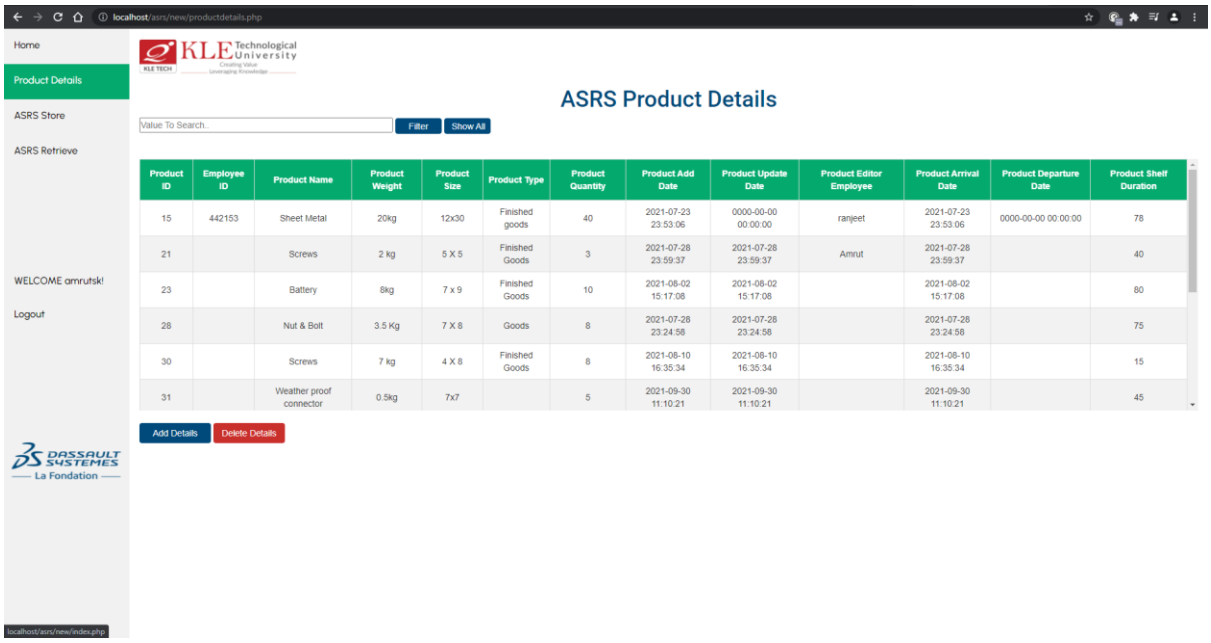
The sidebar allows user to switch between different pages.

The graphs visible are developed in JavaScript. Chartjs provides a simple and efficient way of representing the data visually using JS and HTML5 as various graphs and charts.

The data for the graphs is accessed from the database using PHP and then processed in JS as an array with a little formatting and then fed into the chart input data to get the proper visualization.

Various features like, hovering over the bar, color shifting is provided to the user using Chartjs.

The sizing and formatting is done using CSS.



Product details page:

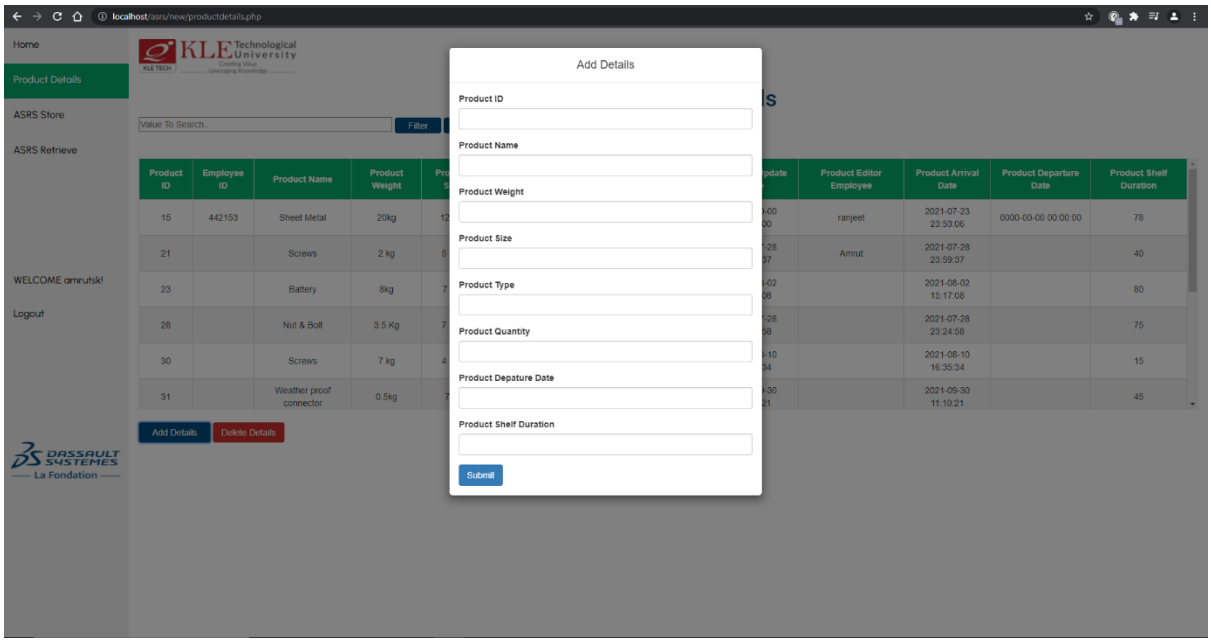
The product details page consists of a table which displays the details of the various products present in the database.

The search bar helps in searching the table of products with any of the desired parameter. The filter button searches the products which satisfy the search value and the Show All button reloads the whole table.

The table is scrollable so that the person can scroll down for further product details.

There are even two buttons to **ADD** a product and **DELETE** a product.

This page uses Bootstrap5 majorly for all these features and style. The formatting is done in HTML5 and style in CSS.



Add Product/Delete Product:

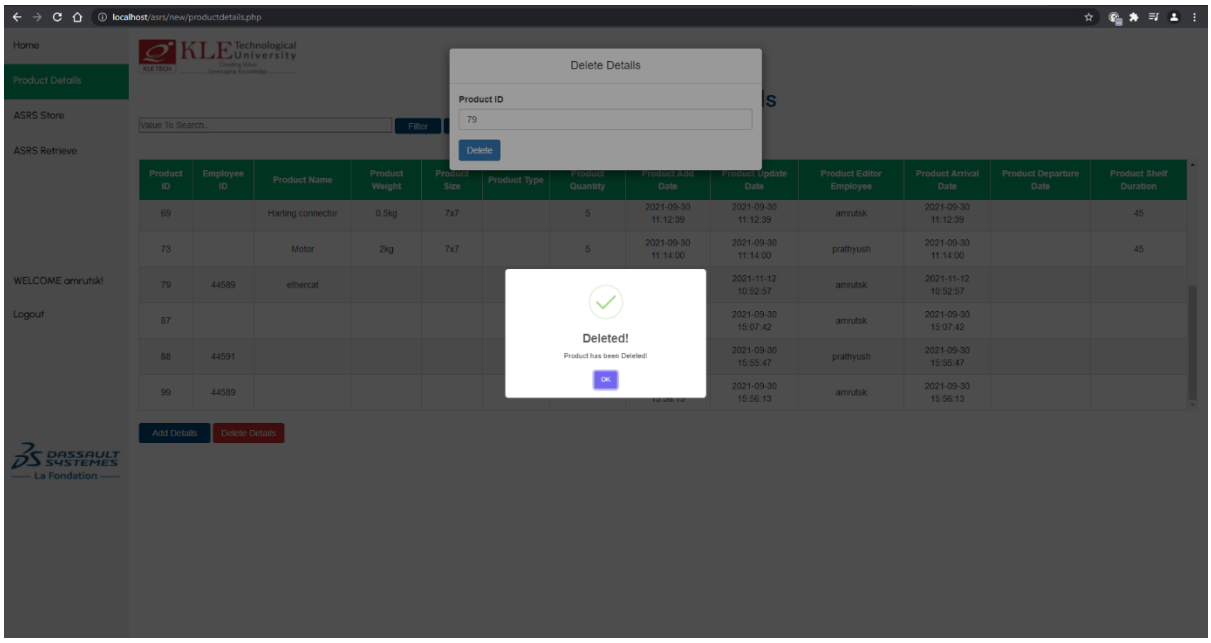
When the buttons is pressed, a pop-up form is obtained, where the user can enter the product details.

The product ID being the primary key of the products table, if the user enters a same Product ID, it shows up a warning.

If none warnings and errors are generated, the product detail is added in the table.

Similarly, the delete product button pops up a form which asks for the product ID to be deleted and deletes the product from the table in the site and also from the database.

Bootstrap5 is used for the pop-up form, its styling and various features like, clicking outside the form closes the form.

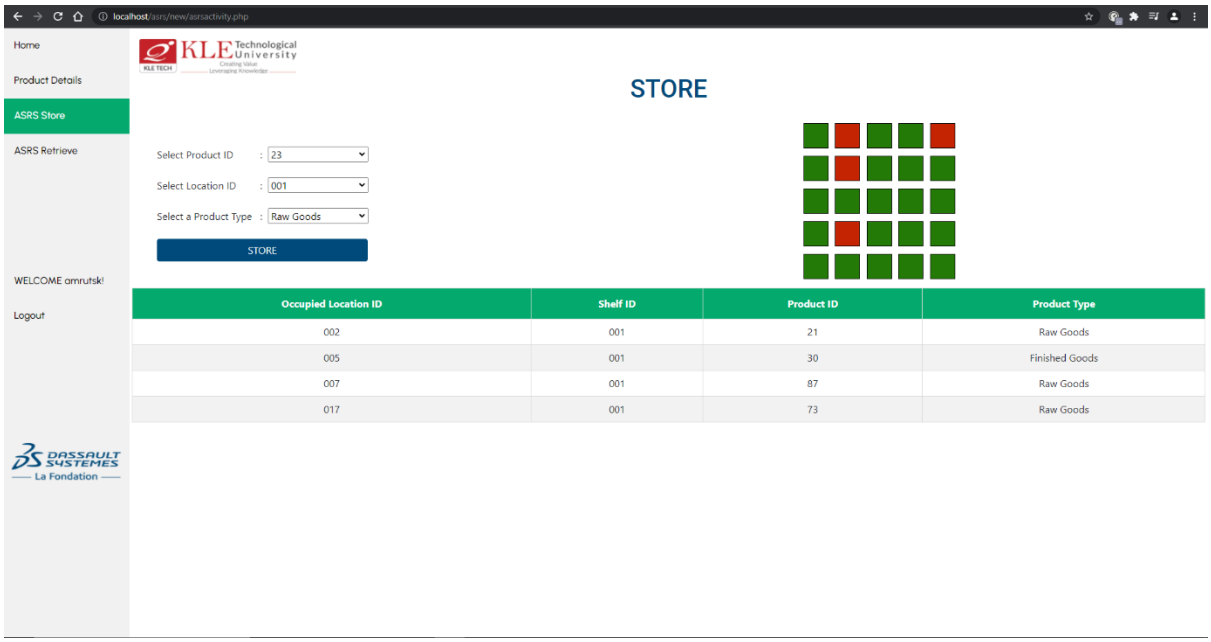


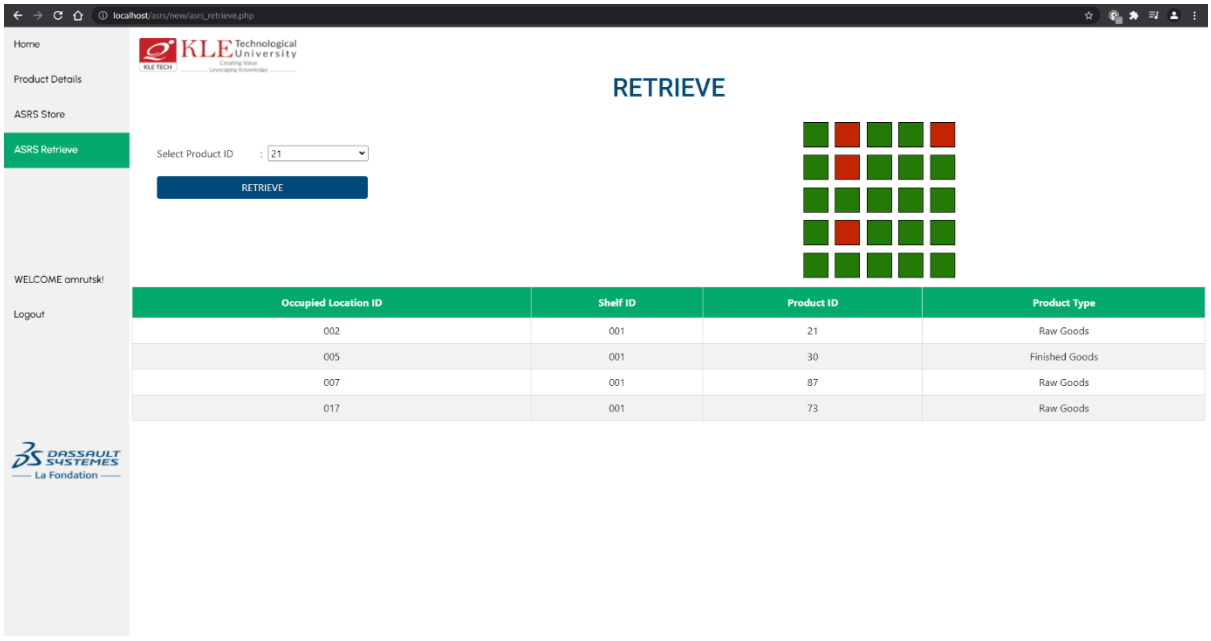
Success/Failure feedback:

After successful addition or deletion of the product, an animated pop-up appears which is been developed using Ajax framework.

Ajax helps in developing web applications or widgets which refresh itself without the whole page being refreshed.

Further application of Ajax is been planned in the site.





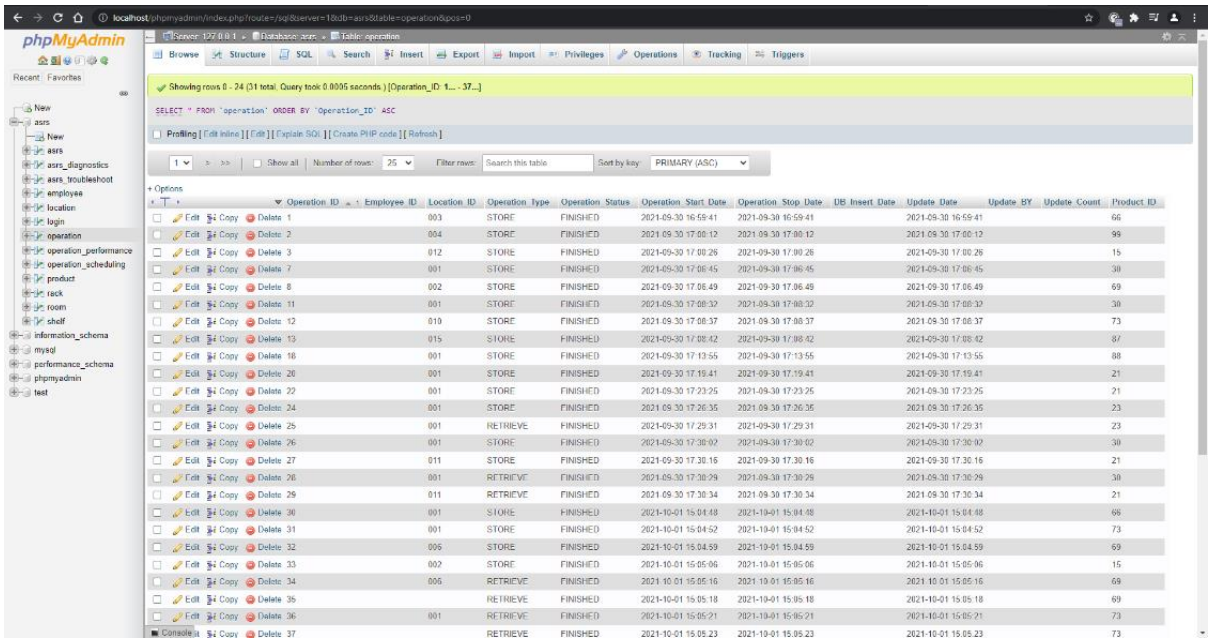
Store and Retrieve page:

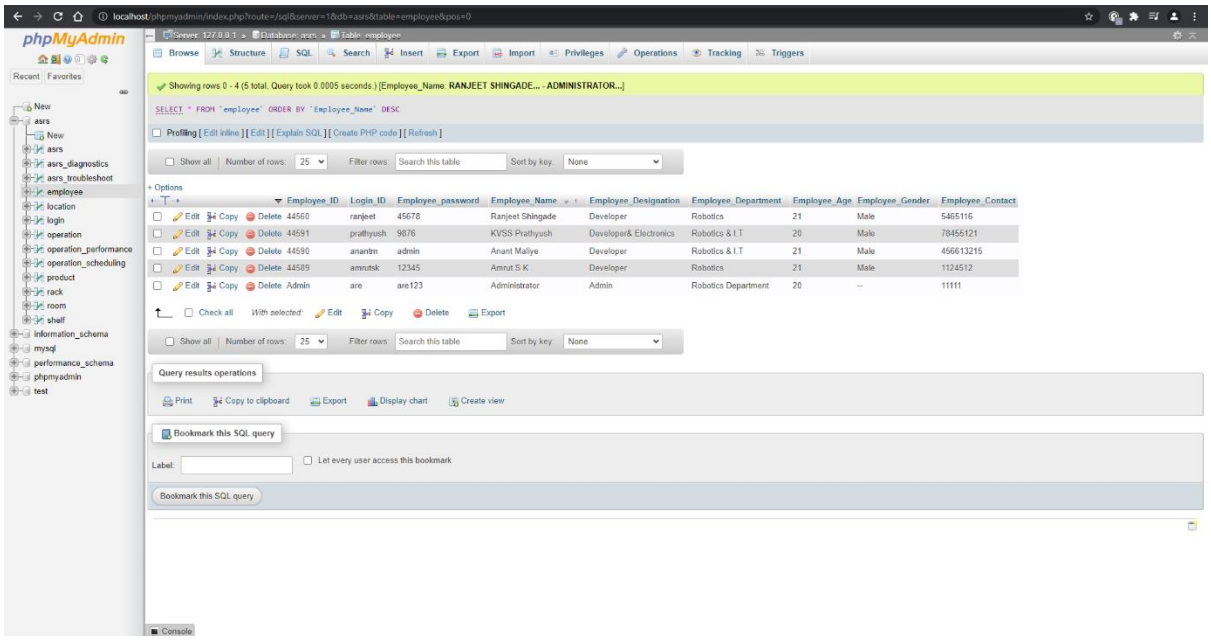
The store and retrieve pages allow the user to either store a product from the product details table into the rack locations.

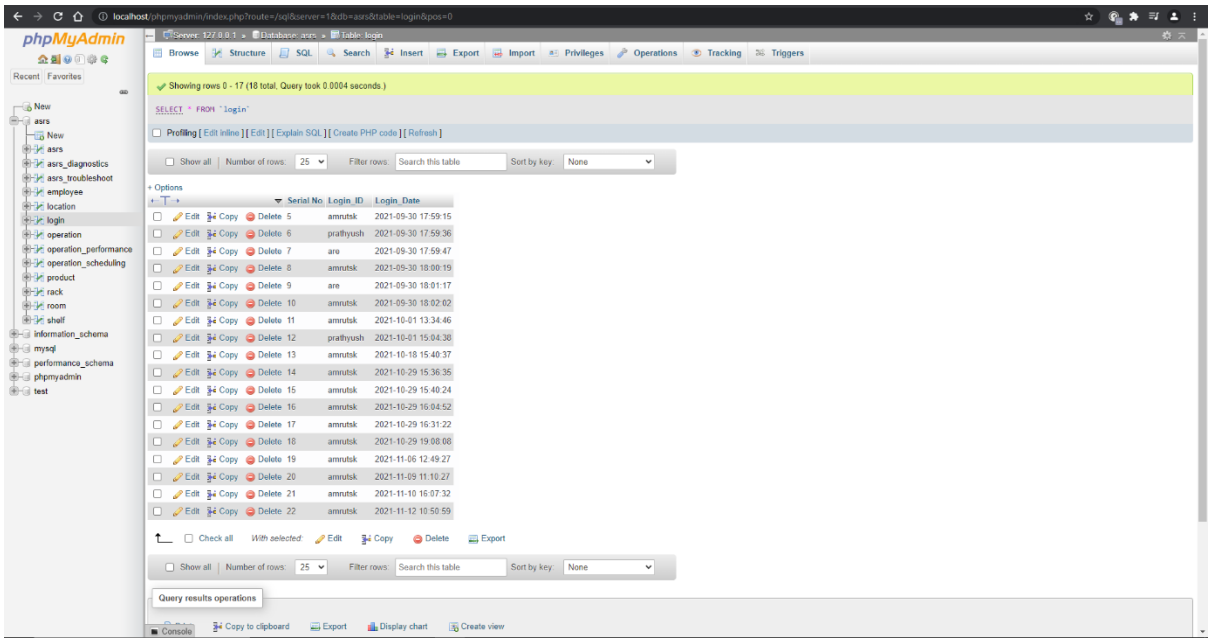
The dropdowns show the storable/retrievable products in these respective pages. The store and retrieve buttons update the table with the particular operations.

The 25 boxes show whether the locations are empty or stored. Green represents an empty location while the Red ones show that the location is occupied with a product which is developed in JS and the details is received from the database using PHP.

4.5 Database


phpMyAdmin
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers
SELECT * FROM `operation` ORDER BY `Operation_ID` ASC
Operation_ID
Employee_ID
Location_ID
Operation_Type
Operation_Status
Operation_Start_Date
Operation_Stop_Date
DB_Insert_Date
Update_Date
Update_BY
Update_Count
Product_ID


phpMyAdmin
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers
SELECT * FROM `employee` ORDER BY `Employee_Name` DESC
Employee_ID
Login_ID
Employee_password
Employee_Name
Employee_Designation
Employee_Department
Employee_Age
Employee_Gender
Employee_Contact
Query results operations
Bookmark this SQL query



The database is been created in MYSQL with PHP as the interface between database and the front-end.

SQL queries were written to create the schema and the tables. The tables are manipulated directly from the site, like insertion and deletion to the tables.

Tables were created as got from the ER diagram.

Tables like:

ASRS

ASRS_Diagnostics

ASRS_Troubleshoot

Employee

Location

Login

Operation

Operation performance

Operation Scheduling

Product

Rack

Room

Shelf

The primary keys and foreign keys have been assigned to the parameters to keep track of the table using it.

Every detail of the ASRS will be stored in the database without any data loss so that a track can be kept on the system.

Ex: who logged in when, how many times, what product was added when, what product was stored at what location and so on.

Chapter 5:

Results

Verification and Validation: -

1. Automatic Storage and Retrieval

   This test case was taken into account for completely automated storage and retrieval procedures of the ASRS.

   One – cycle time for Min. location [RETREIVAL**]** = 1min: 57 sec

   One – cycle time for Max. location [RETREIVAL**]** = 2min: 21 sec

   One – cycle time for Min. location [STORAGE] = 26 sec

   One – cycle time for Max. location [STORAGE] = 1min: 23 sec

   Database update time lag = < 1 sec

2. Manual Storage and Retrieval

   This test case was taken into account for completely manual storage and retrieval procedures of the ASRS with the help of HMI designed in Twincat and Web based User Interface.
   One – cycle time for Min. location [RETREIVAL] = 1min: 57 sec

   One – cycle time for Max. location [RETREIVAL] = 2min: 21 sec

   One – cycle time for Min. location [STORAGE] = 26 sec

   One – cycle time for Max. location [STORAGE] = 1min: 23 sec

   Database update time lag = < 1sec

   UI and actuation lag = < 1sec

3. OPC server connectivity and Database Update

   This test case was taken into account to verify and validate the updating of the Backend established Database for proper registration of the values into database upon completion of one cycle of defined case and connectivity between between ASRS – TwinCAT and UI and MySQL database with the help of OPC – UA server which has been established.
   Parameters to be checked: -

   - Successful updating of Database with Storage location allocated.
   - Successful updating of Database for Retrieving the object order received.

4. Testing a whole Automation cycle with integrated conveyor

   This test case was taken into account for testing a complete automation process designed which involved an integrated conveyor system and taking a payload or box and automatically allocating a storage location.

   One Complete cycle time = 2min: 15 sec

   Time taken for retrieving empty storage location from database = < 1 sec

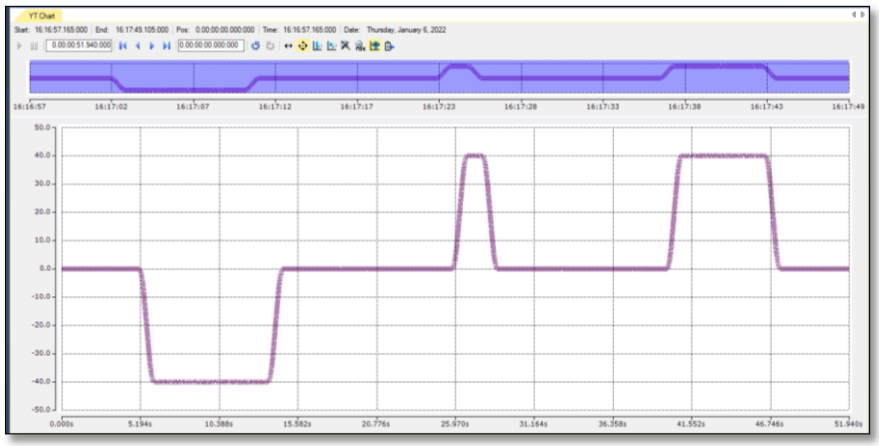
   Database update time lag upon completion of cycle = < 1sec

5. Integration of ASRS system with Digital Twin [MATLAB]

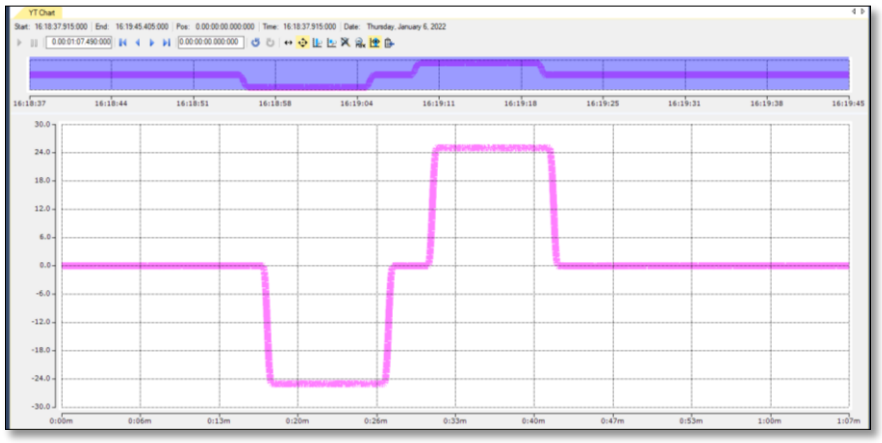
   This test case was taken into account for testing the digital twin designed in MATLAB with the integration of real time operation of ASRS with the help of TwinCAT OPC – UA server established for connectivity.

   Time delay between actual run and digital twin simulation = 2 – 3 sec

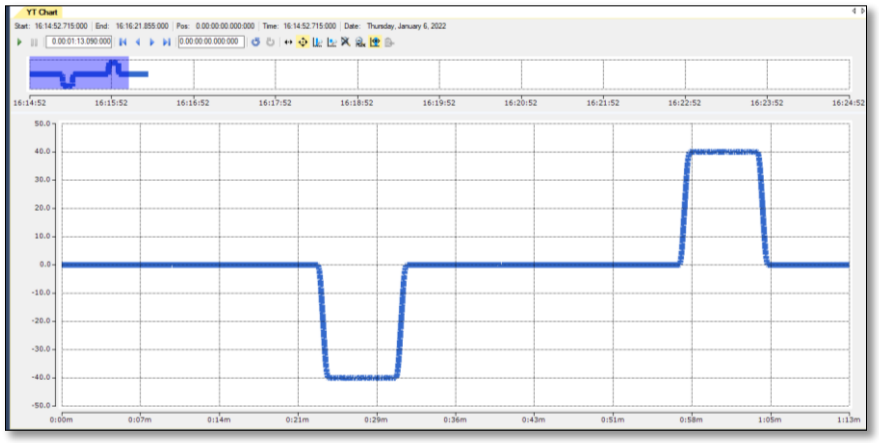
Twincat Scope Graphs:



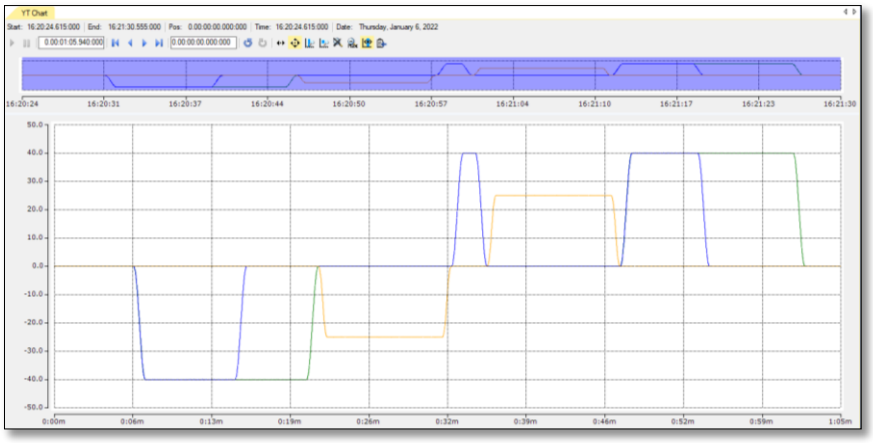
Y Axis Store 6th Location



Z Axis Store 7th Location



X Axis Store 2nd Location



XYZ Storage 8th Location

Chapter 6:

Conclusion

This report summarizes our attempt at designing an Automated Storage and Retrieval System. The most optimum design and methodologies have been presented here. Results were also compared with material selection table and software analysis. This can be further used to make working models of what is now on paper. Variations in the design as per requirement can be done, with proper implementation of mechanical and electronic components, variety of automation can be achieved, which can be as complex, versatile and flexible as required. This model can serve as a reference for various applications, some of which are automatic car parking system, automatic baggage handling for loading/unloading at airports, inventory and raw material management in large industries, automated parcel sorting in postal services, books management in huge libraries, etc.

Chapter 7:

Future Scope

The proposed prototype is preliminary model of industry level automatic storage and retrieval system. Several new and innovative features can be added to the present version to make it stand out from the existing solutions.

Below listed are few of such features:

- The UI can be developed in such a way, for the user to schedule various store and retrieval processes at particular time without collision of commands
- The digital twin can be integrated with the site developed for the ASRS and displayed there, real-time, using GStreamer libraries; reducing the number of screens required for monitoring the complete system.
- A QR code can be assigned to each box and a QR code scanner (camera) can be installed on the system to track the box in real time and act as the feedback for the whole system.
- The prompt in the ASRS store and retrieve pages can be developed as a console for the user to add commands in (Assembly level language) and to display the debug, directly to and from the TwinCat platform.
- The scope data of the motors can directly be taken into the digital twin, and a model for predictive maintenance can be developed.
- The data of the operations undertaken by the system can be used to learn the trends of the incoming and outgoing products, like their shelf life, their frequency and the locations they can be stored for greater efficiency of the system.
- Guideways can be installed on the conveyor system, the z-axis or the racks itself, to always keep the boxes in proper orientation and avoiding the system from collapsing onto each other.
- The site can be tweaked in order to provide each user with different versions of the site, say operator , supervisor or manager, with limited set of their functions, so that no other person can misuse or mishandle the ASRS.
- Various conveyor systems can be integrated together with the pick and place robot to send and receive the goods from various stations and showcase the execution of the Industry 4.0

Appendix 1

Components Used with Their Specifications (Electrical)

1. C6030-0070 / 000131560 / Ultra Compact Control Cabinet IPC :

The Industrial PC sector from Beckhoff offers superior hardware solutions for every requirement. Whether Panel PCs, PCs, Control Panels or Embedded PCs. Learn more about the advantages of IPC technology from Beckhoff. It has Intel® Celeron® G4900 3.1 GHz, 2 cores (TC3: 50) processor. It has compact motherboard for 8th/9th generation Intel® Celeron®, Pentium®, Core™ i3/i5/i7.

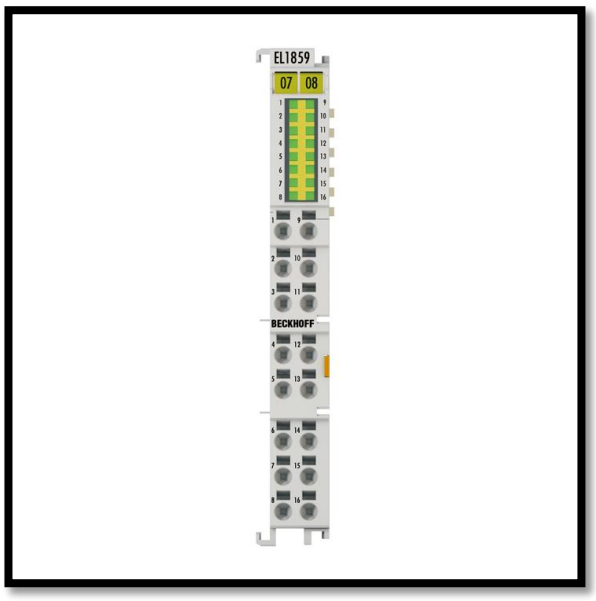
2. EK1100 / Ether CAT Coupler for E-Bus-terminal(ELxxxx)

The EK1100 EtherCAT Coupler is the link between the EtherCAT protocol at fieldbus level and the EtherCAT Terminals. The coupler converts the passing telegrams from Ethernet 100BASE-TX to E-bus signal representation. A station consists of a coupler and any number of EtherCAT Terminals that are automatically detected and individually displayed in the process image.

Special features:

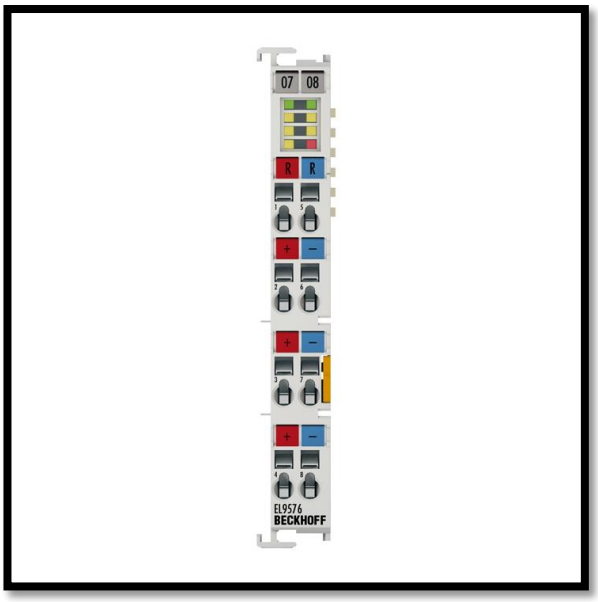
- Connection technology: 2 x RJ45 socket
- Connection lengths: up to 100 m
- Number of EtherCAT Terminals in the overall system: up to 65,535

3. EL1859 / 8-channel digital input and output terminal



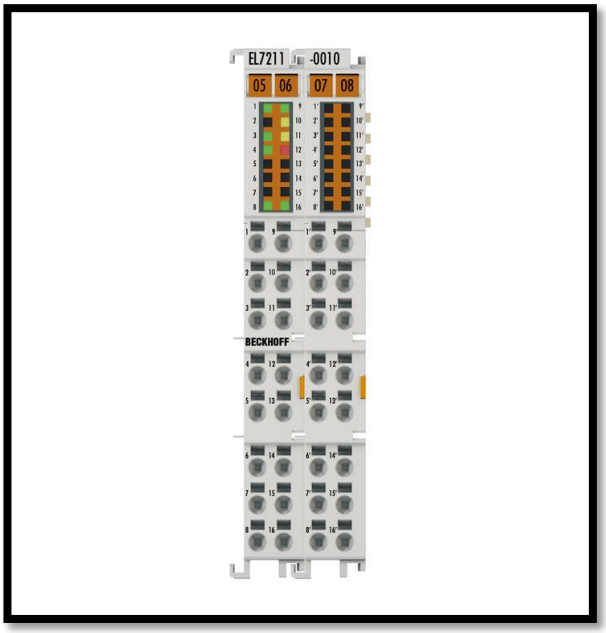
The digital EL1859 EtherCAT Terminal combines eight digital inputs and eight digital outputs on a single device. The signal state of the channels is indicated by LEDs. The reference ground for all inputs is the 0 V DC power contact; the outputs are fed via the 24 V power contact.

4. EL9576 / Brake chopper terminal



The EL9576 EtherCAT Terminal contains high-performance capacitors for stabilising supply voltages. The EL9576 can be used in conjunction with the EL70x1 stepper motor terminals, the EL73x2 DC motor terminals and the EL72x1 servomotor terminals. Low internal resistance and high-pulsed current capability enable good buffering in parallel with a power supply unit.

5. EL7211-0010 / Servomotor terminal with OCT integrated



The EL7211-0010 servomotor EtherCAT Terminal with integrated One Cable Technology (OCT) offers high servo performance in a very compact design for motors from the AM8100 up to 4.5 A (Irms). The One Cable Technology combines a motor cable and an absolute feedback system in a single cable.

6. ZB8110 / External ballast resistor 10 Ohm, 100 W

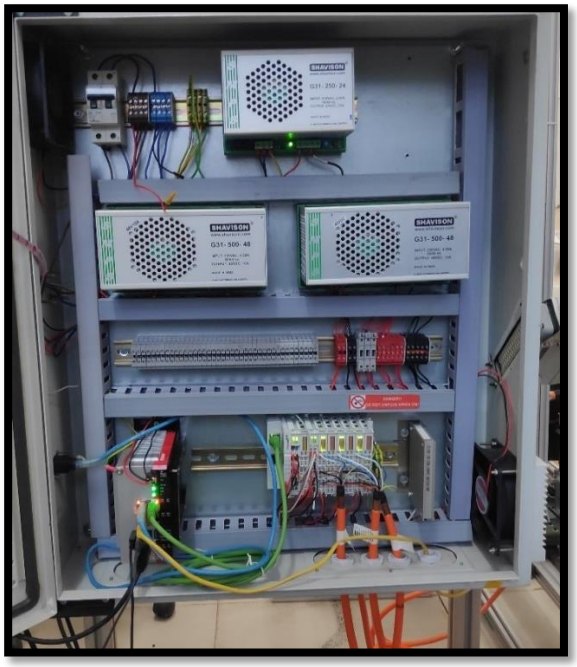
During the acceleration phase, a motor needs energy supply, whereas during braking a motor can be used as a generator, feeding energy back into the DC-Link, which raises the voltage in the DC-Link. If the voltage exceeds the adjustable threshold value, a braking resistor is activated.

The external ZB8110 braking resistor is available as an accessory to the EL9576 brake chopper terminal or the KL9570 buffer capacitor terminal. It regulates the DC-Link voltage as soon as more braking power is needed. It has a maximum continuous rating of 100 W. The ZB8110 is connected directly to the EL9576 or KL9570. A mounting plate is included in the scope of supply for each braking resistor.

7. AM8112-0F21-0000 / 000009147

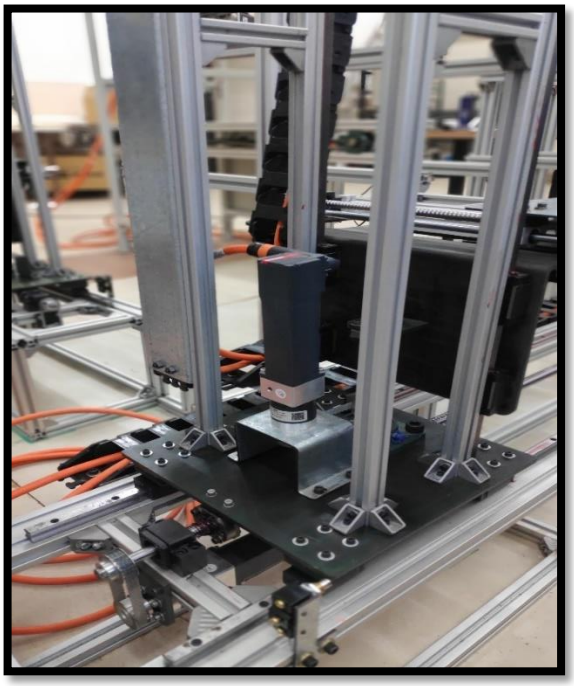
The AM8112 low-voltage servomotor is suitable for highly dynamic drive solutions in compact design in the voltage range of < 50 V DC. The standstill torque of the motor depends on the winding and is in the range of < 1.5 Nm. It is available with the OCT feedback system (absolute encoder). The low-voltage servomotor with flange code F1 (40 mm) and motor length 2 has a shaft diameter b = 8 h7 and a free shaft end of d = 25 mm.

GALLERY



Control Panel

Figure 1 ASRS Assembly



X- Axis of ASRS

Y- Axis of ASRS

Z - Axis of ASRS

Appendix 2

Bill Of Materials (Mechanical)

SL.NO	PRODUCT DESCRIPTION	SPECIFICATION	QTY.	UNITS	PART NO.
Ball Screw					
1	Ball Screw	16mm diameter, Pitch = 5mm Length= 1710mm(x-axis)	2	pc	--
		16mm diameter, Pitch = 5mm Length= 1453mm(y-axis)	2	pc	--
		12mm diameter, Pitch = 5mm Length= 455mm(z-axis)	2	pc	--
Ball Screw Nut					
2	Nut	16mm diameter(x-axis)	2	pc	--
		16mm diameter(y-axis)	2	pc	--
		12mm diameter(z-axis)	2	pc	--
Bearings					
3	Linear Bearing	12mm ID, 24mm OD	8	pcs	61901
	4 bolt flange Bearing	15mm ID	4	pcs	FY12
	Linear Rail Shaft Support	8mm diameter	4	pcs	SK8
	Linear Bearing	8mm Inner diameter	4	pcs	SC8UU
	LM Bearing LM 10UU	10mm Inner diameter	8	pcs	--
	Ballscrew end supporter	10mm diameter (z-axis)	4	pcs	BF12
		12mm diameter (x-axis)	2	pcs	BF12
		12mm diameter (x-axis)	2	pcs	BK12
LM Guideway Block					
4	Guide Block	Width=25mm(x and y-axis)	16	pcs	HGH15CAZ0C
5	LM Guides	Width=15mm, Length=1400mm (y-axis)	4	pcs	HGR25R1800C
	Guideway	Width=25mm, Length=1800mm (x-axis)	4	pcs	HGR15R1400C

Structures					
6	Structures Aluminium extrusions(Bosch)	**30x30**	--	feet	--
		X axis 240mm	8	pcs	--
		235mm	16	pcs	--
		2080mm	4	pcs	--
		180mm	6	pcs	--
		Rack 200mm	4	pcs	--
		430mm	4	pcs	--
		1430mm	4	pcs	--
		2000mm	24	pcs	--
		1490mm	4	pcs	--
		260mm	100	pcs	--
		20x20	--	feet	--
		Y axis 1490mm	8	pcs	--
		100mm	12	pcs	--
		220mm	4	pcs	--
		Z axis 490mm	4	pcs	--
		112mm	4	pcs	--
		60x30			--
		x axis 2080mm	4	pcs	--
Timing Belt - Pulley					
7	Pulley	Teeth=20, width=10mm, ID=12mm(x-axis & y-axis)	4	pcs	--
		Teeth=20, width=10mm, ID=10mm(x-axis & y-axis)	4	pcs	--
		Teeth=20, width=10mm, ID=10mm(z-axis)	4	pcs	--
8	Timing Belt	loop length=200mm,width=10mm(y-axis)	2	pcs	--
		loop length=315mm,width=10mm(x-axis)	2	pcs	--
		loop length=315mm,width=10mm(z-axis)	2	pcs	--

Sheet Metal					
9	MS sheet metal	**X-axis**			
		Thickness = 6mm, lengthx breadth=800x800mm	1	pcs	--
		Y & Z-axis			--
		Thickness = 5mm, length x breadth=1000x1000mm	1	pcs	--
	GI sheet metal	**X,Y & Z axis**			
9		MS sheet metal	Thickness = 3mm , length x breadth=1000x2000mm **X-axis**	1	pcs
Fasteners, Nuts and Washers					
10	Allen Bolt	M3x8	20	pcs	--
		M3x12	10	pcs	--
		M4x10	20	pcs	--
		M4X12	30	pcs	--
		M4x20	30	pcs	--
		M4x25	30	pcs	--
		M4x30	30	pcs	--
		M4x35	150	pcs	--
		M5x12	20	pcs	--
		M5x16	10	pcs	--
		M5x20	10	pcs	--
		M5x45	40	pcs	--
		M6x30	10	pcs	--
		M6x40	10	pcs	--
11	Spring Washers	M3	30	pcs	--
		M4	290	pcs	--
		M5	80	pcs	--
		M6	20	pcs	--
12	Washers	M3	30	pcs	--
		M4	290	pcs	--
		M5	80	pcs	--
		M6	20	pcs	--
13	T nut and bolt				

		20x20	35	pcs	--
14	Gussets	30x30	65	pcs	--
		38x38	44	pcs	--
Guide Rod					
15	**z-axis**	460mm	2	pcs	--
		510mm	2	pcs	--
Track chain					
16	**x-axis**	1200mm	2	pcs	--
	y-axis	1200mm	2	pcs	--
Shaft end support					
	z-axis	Shaft end support	8	pcs	SK10

Bill Of Materials (Electrical)

ITEM. NO	PART NUMBER	QTY.	VENDOR
1	Shavison G31-250-24	1	SHAVISON ELECTRONICS PVT. LTD.
2	Shavison G31-500-48	2	
3	Meanwell 24v SMPS	1	MEAN WELL Enterprises Co
4	Meanwell 48v SMPS	2	
5	MCB 2p	2	Prajay Electrotech, Hubli
6	Terminal blocks	{4L ,4N, 4G, 24v(6p,4n),48v(2p, 2n), Data blocks(30)}x2	
7	Indicator	2	
8	Toggle on/off Power Switch	2	
9	Emergency Switch	2	
10	Ethernet port	2	
11	Exhaust Fans	4	
12	Harting Connector (32pins)	2	
13	Weather proof connector	2	
14	Fuse 10A	6	
15	Wire duct and Dendrill	2 set	
16	Limit switch – 15A, 250VAC [ID15GQ-B]	12	

BECKOFF Electrical Components			
17	C6030-0070 (IPC)	1	Beckhoff Automation
18	EL1859 8 channel digital I/O terminal	2	
19	EL9576 Brake chopper terminal	2	
20	EL7211-0010 Servo motor terminal	6	
21	El9011 Bus end cap	2	
22	ZB8110 External ballast Resistor (10R, 100W, 600Vac)	2	
23	AM8112-0F21-0000 Motor	6	
24	ZK1090-9191-0000 Ethernet patch Cable	2	
25	Ek1100 EtherCAT terminal for E-bus terminal	2	
26	ZK4704-0421-2050 Motor power cable	6	

Chapter 8:

REFERENCES

[1] Roodbergen, K.J. and Vis, I.F.A. (2009), A survey of literature on automated storage and retrieval systems. European Journal of Operational Research 194(2), 343-362

[2] Design optimization and development of an Automated Storage and Retrieval System, T Ajay et al 2020 IOP.

[3] Design and Development of Prototype for Automated Storage and Retrieval System, Tayyaba Mudasser, Qurat-Ul-Ain.

[4] Barbato, Michele & Ceselli, Alberto & Premoli, Marco. (2019). Evaluating Automated Storage and Retrieval System Policies with Simulation and Optimization. 127-137. 10.1007/978-3-030-34960-8_12.

[5] Roodbergen, Kees Jan and Iris F. A. Vis. "A survey of literature on automated storage and retrieval systems." Eur. J. Oper. Res. 194 (2009): 343-362.

[6] Mohammad Khasasi, Farah Hanani & Mohd Yusof, Zulkhairi & Alias, Mohd & Adam, Ismail & Rasin, Mohamad. (2016). Development of Automated Storage and Retrieval System (ASRS) for Flexible Manufacturing System (FMS). Journal of Engineering Technology 2231-8798. 4. 43-50

[7] Ullman, David & Dietterich, Thomas. (1987). Mechanical design methodology: Implications on future developments of Computer-Aided Design and Knowledge-Based Systems. Engineering with Computers. 2. 21-29. 10.1007/BF01200174.

[8] Vahid, Salehi & Wang, Shirui. (2017). USING POINT CLOUD TECHNOLOGY FOR PROCESS SIMULATION IN THE CONTEXT OF DIGITAL FACTORY BASED ON A SYSTEMS ENGINEERING INTEGRATED APPROACH.

[9] Abdel Gawad, Ahmed. (2015). Multidisciplinary Engineering for the Utilization of Traditional Automated Storage and Retrieval System (ASRS) for Firefighting in Warehouses. American Journal of Energy Engineering. 3. 1-22. 10.11648/j.ajee. s.2015030401.11.