Connect TwinCAT 4026 with Transport and Sorting Line Model.

In this chapter we are going to develop the initial logic needed to map the Transport and Sorting Line model with TwinCat 4026.

First we need to understand which Inputs and Outputs are contained in the model.

An input is a sensor which will indicate to the system something is happening. Output tell the devices or actuators what to do, base of these two components is how the automation perform the task.

They can be different types of inputs for example:

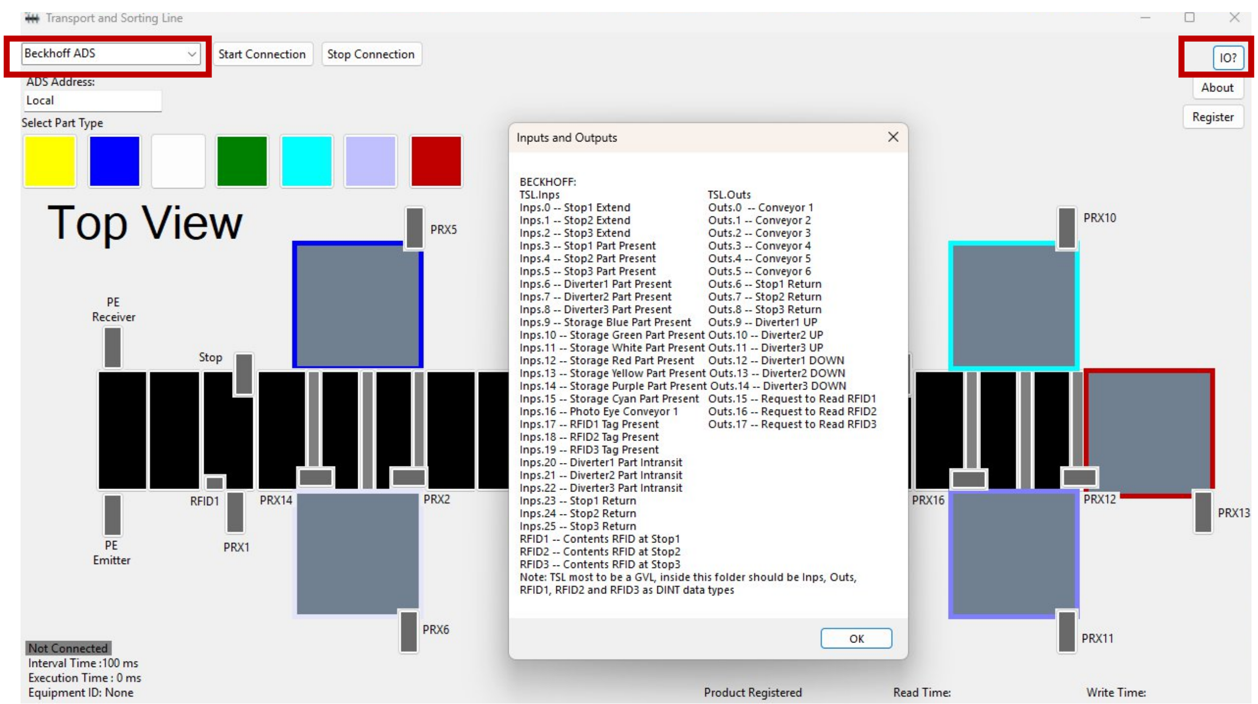
Proximity sensor, push button switch, vacuum sensor, photo eye sensor, light curtain, pick light, etc...

Exist multiple kind of sensors, outputs or actuators have multiple types of devices, for example:

Motor, Drives, Valves, Lights, Relays, etc.....

We will maintain it simple. Automation System can be very complicate. The purpose of these training is to learn to develop logic using SCL (structure control language) on TwinCAT 4026. No need to complicate it even more.

In order to identify the inputs and output on the model, we need to select Beckhoff ADS and then click in the bottom which say IO. Beckhoff ADS is the communication path that the application use to connect with our TwinCAT 4026 application.



We can see that the model contains 26 digital inputs, 18 digital outputs and three double integers defined as RFID data.

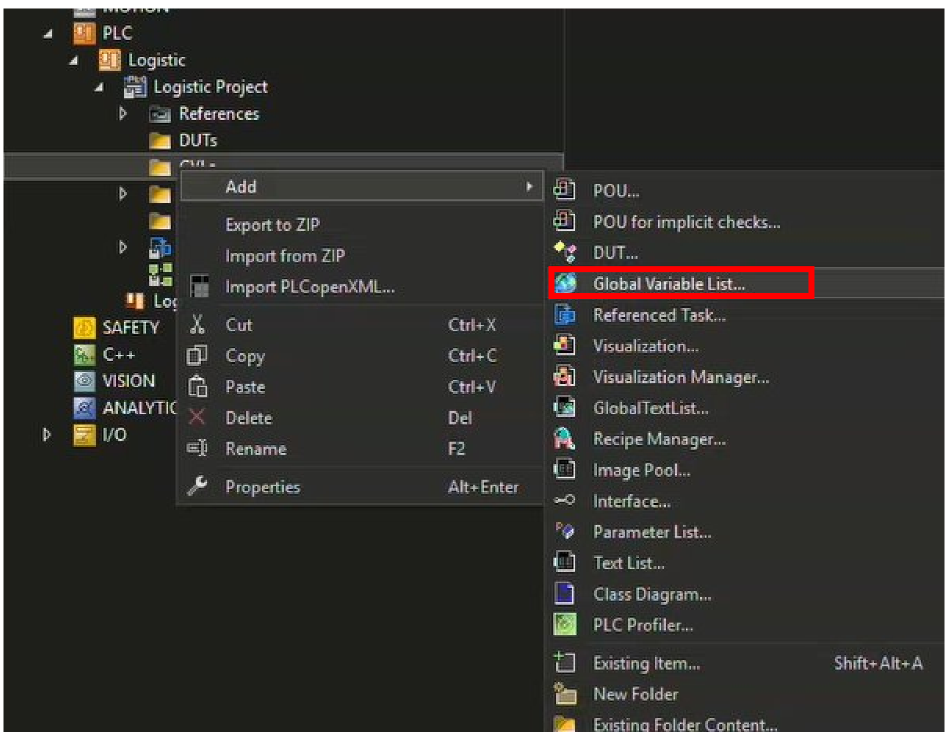
Table for the inputs at Global Variable List TSL:

Input address	Description
Inps.0	Stop1_Extend
Inps.1	Stop2_Extned
Inps.2	Stop3_Extend
Inps.3	Stop1_PartPresent
Inps.4	Stop2_PartPresent
Inps.5	Stop3_PartPresent
Inps.6	Diverter1_PartPresent
Inps.7	Diverter2_PartPresent
Inps.8	Diverter3_PartPresent
Inps.9	Storage_Blue_PartPresent
Inps.10	Storage_Green_PartPresent
Inps.11	Storage_White_PartPresent
Inps.12	Storage_Red_PartPresent
Inps.13	Storage_Yellow_PartPresent
Inps.14	Storage_Purple_PartPresent
Inps.15	Storage_Cyan_PartPresent
Inps.16	Photo_Eye_Conveyor1
Inps.17	RFID1_Tag_Present
Inps.18	RFID2_Tag_Present
Inps.19	RFID3_Tag_Present
Inps.20	Diverter1_Part_Intransit
Inps.21	Diverter2_Part_Intransit
Inps.22	Diverter3_Part_Intransit
Inps.23	Stop1_Return
Inps.24	Stop2_Return
Inps.25	Stop3_Return
RFID1_Data	Information contained at RFID1
RFID2_Data	Information contained at RFID 2
RFID3_Data	Information contained at RFID 3

Table for the Outputs at Global Variable List TSL:

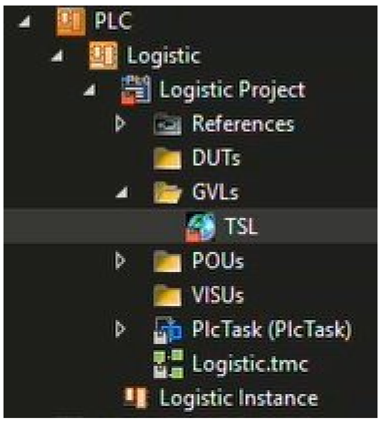
Output address	Description
Outs.0	Conveyor1
Outs.1	Conveyor2
Outs.2	Conveyor3
Outs.3	Conveyor4
Outs.4	Conveyor5
Outs.5	Conveyor6
Outs.6	Stop1_Return
Outs.7	Stop2_Return
Outs.8	Stop3_Return
Outs.9	Diverter1_UP
Outs.10	Diverter2_UP
Outs.11	Diverter3_UP
Outs.12	Diverter1_Down
Outs.13	Diverter2_Down
Outs.14	Diverter3_Down
Outs.15	Req_RFID1
Outs.16	Req_RFID2
Outs.17	Req_RFID3

Now that have are all variables defined. First we are going to create a global variable list called TSL to map the elemental variables, which are Inps, Outs, RFID1, RFID2 and RFID3.

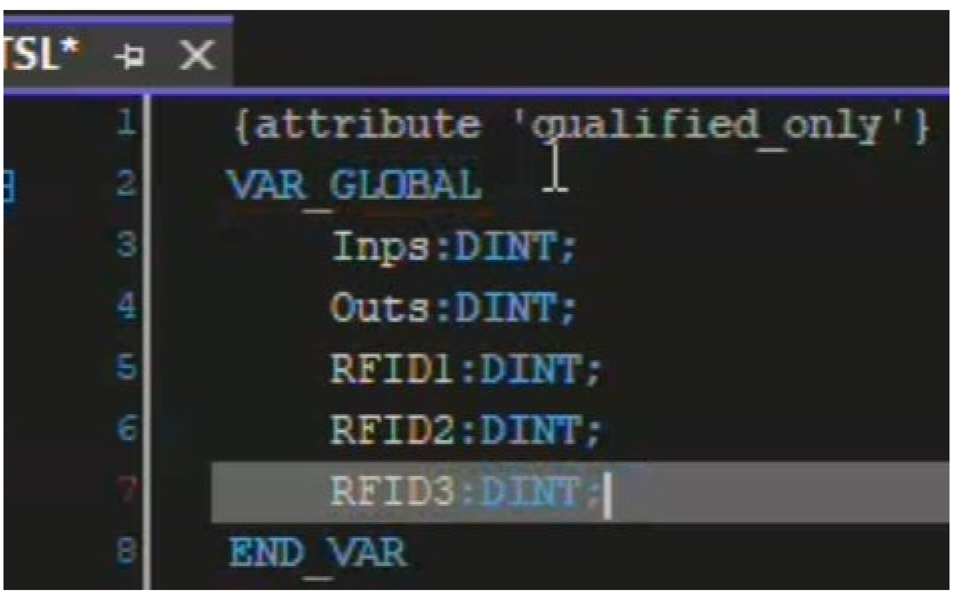


TSL most to be the name. If we do not name it TSL, the model will not be able to communicate to TwinCAT Runtime (TwinCAT Runtime is the peace of software which execute the different task of the platform, for example PLC code, Inputs and Outputs communication).

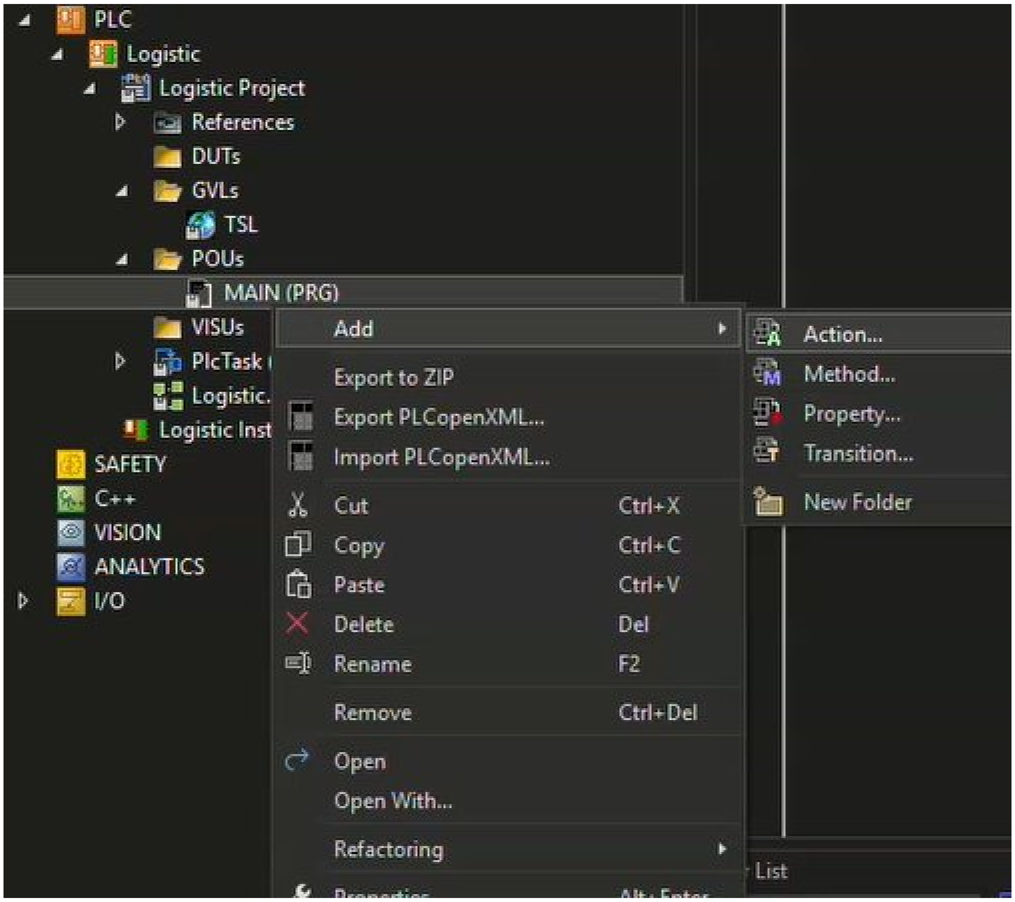
Once the new Global Variable is created. We can open it in order to add the variables.



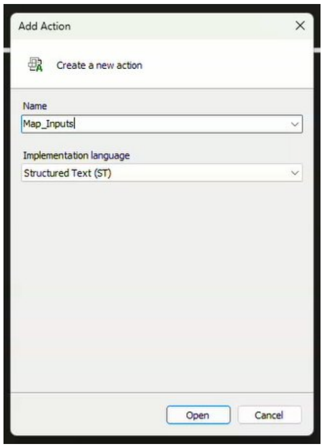
Inside TSL GVL (Global Variable List), we need to add the next variables Inps, Outs, RFID1, RFID2 and RFID3. Variables declaration require to use the VarName : DataType; format.



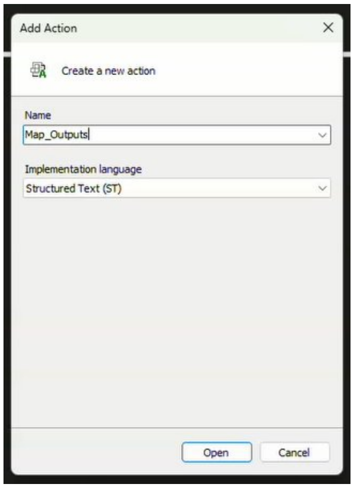
We are going to create two actions on the main program. Map_Inputs and Map_Outputs.We will use these two to map the digital value of our inputs and outputs respectively.



We are going to call the first one Map_Inputs.

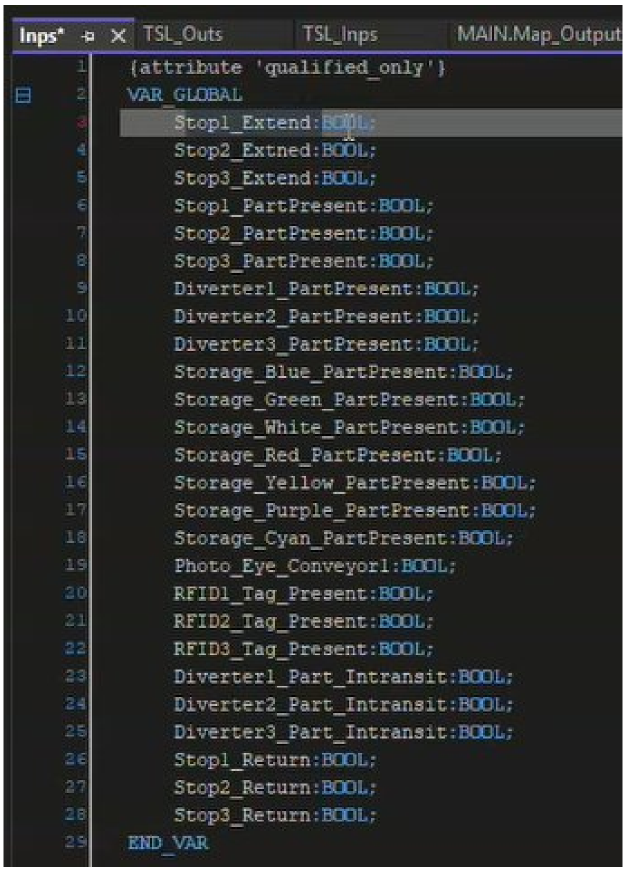


And second one Map_Outputs

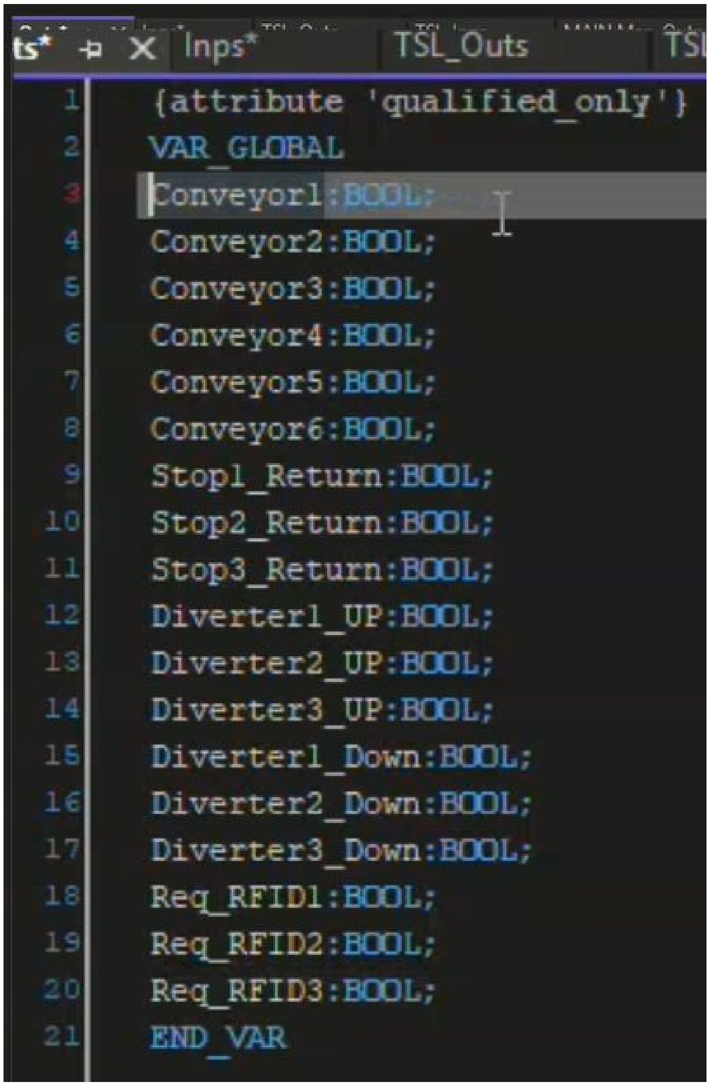


Now we are going to create another GVL (Global Variable List) called Inps. In this Global Variable List we are going to create the map of our inputs variables which will be more user friendly to use instead of Inps.X or Outs.x from TSL GVL. These memory assignation will help to write the program.

Global Variable List name Inps variables contain the signals of the table above. We can see that follow the same partner Var_Name: DataType; In these case the great majority of variables are bool, with the exception of RFID Data.

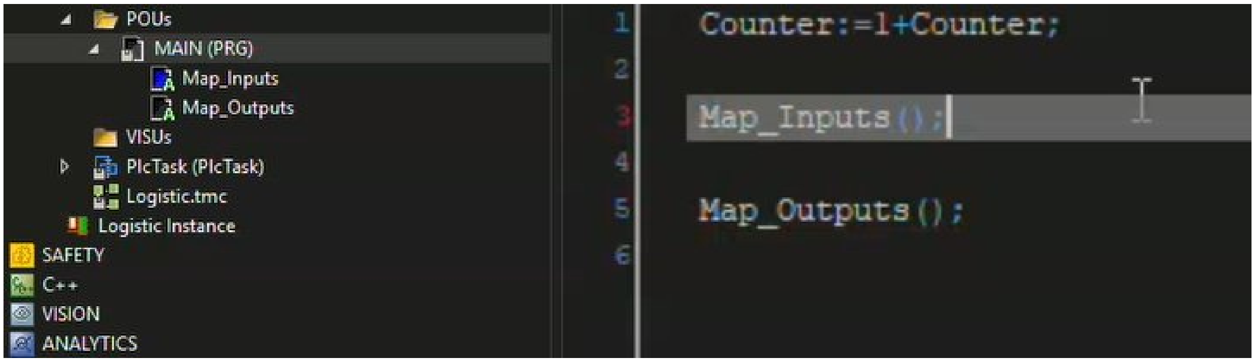


We need create another GVL called Outs. In this GVL we are going to create the map of our Outputs variables, which we are going to use to automate our process. All these variables are boolean.

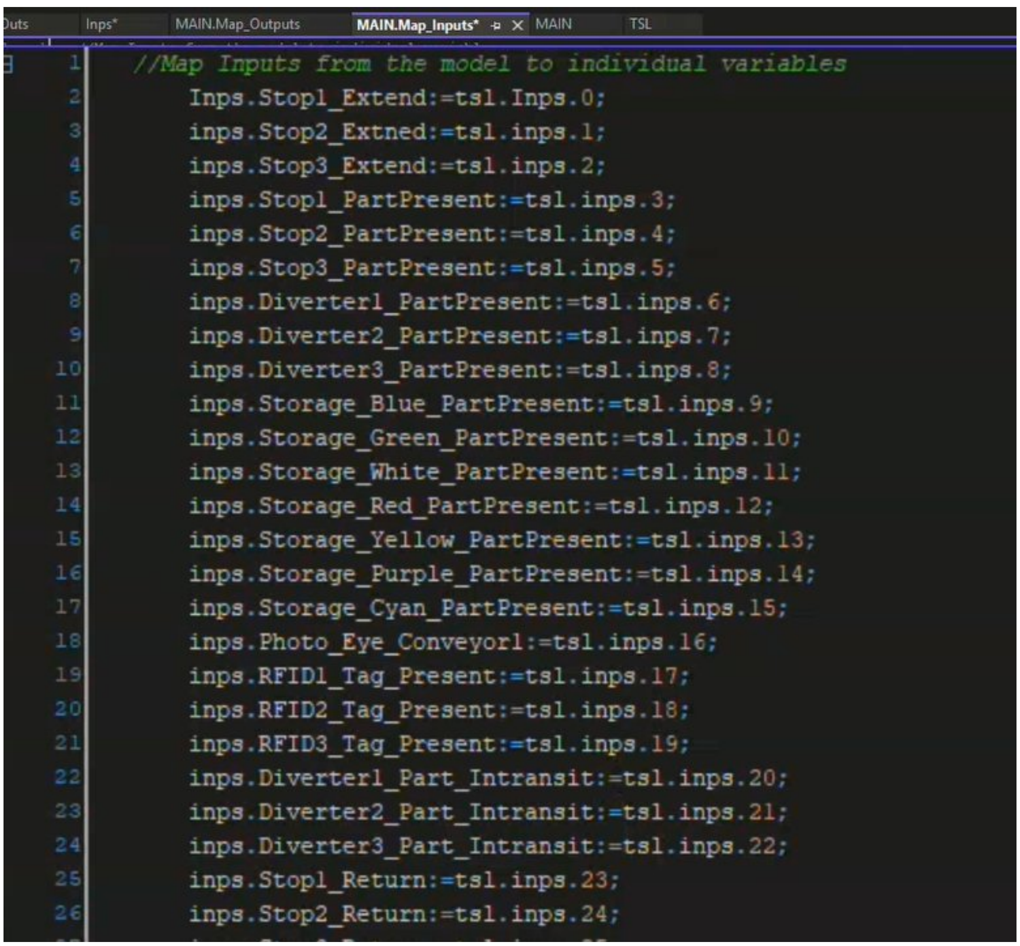



```
{attribute 'qualified_only'}
VAR_GLOBAL
Conveyor1:BOOL;
Conveyor2:BOOL;
Conveyor3:BOOL;
Conveyor4:BOOL;
Conveyor5:BOOL;
Conveyor6:BOOL;
Stop1_Return:BOOL;
Stop2_Return:BOOL;
Stop3_Return:BOOL;
Diverter1_UP:BOOL;
Diverter2_UP:BOOL;
Diverter3_UP:BOOL;
Diverter1_Down:BOOL;
Diverter2_Down:BOOL;
Diverter3_Down:BOOL;
Req_RFID1:BOOL;
Req_RFID2:BOOL;
Req_RFID3:BOOL;
END_VAR
```

Now we are going to call the actions which we created already. The actions are part of the main program, but we need to call them in order to be executed. We called on the main program:

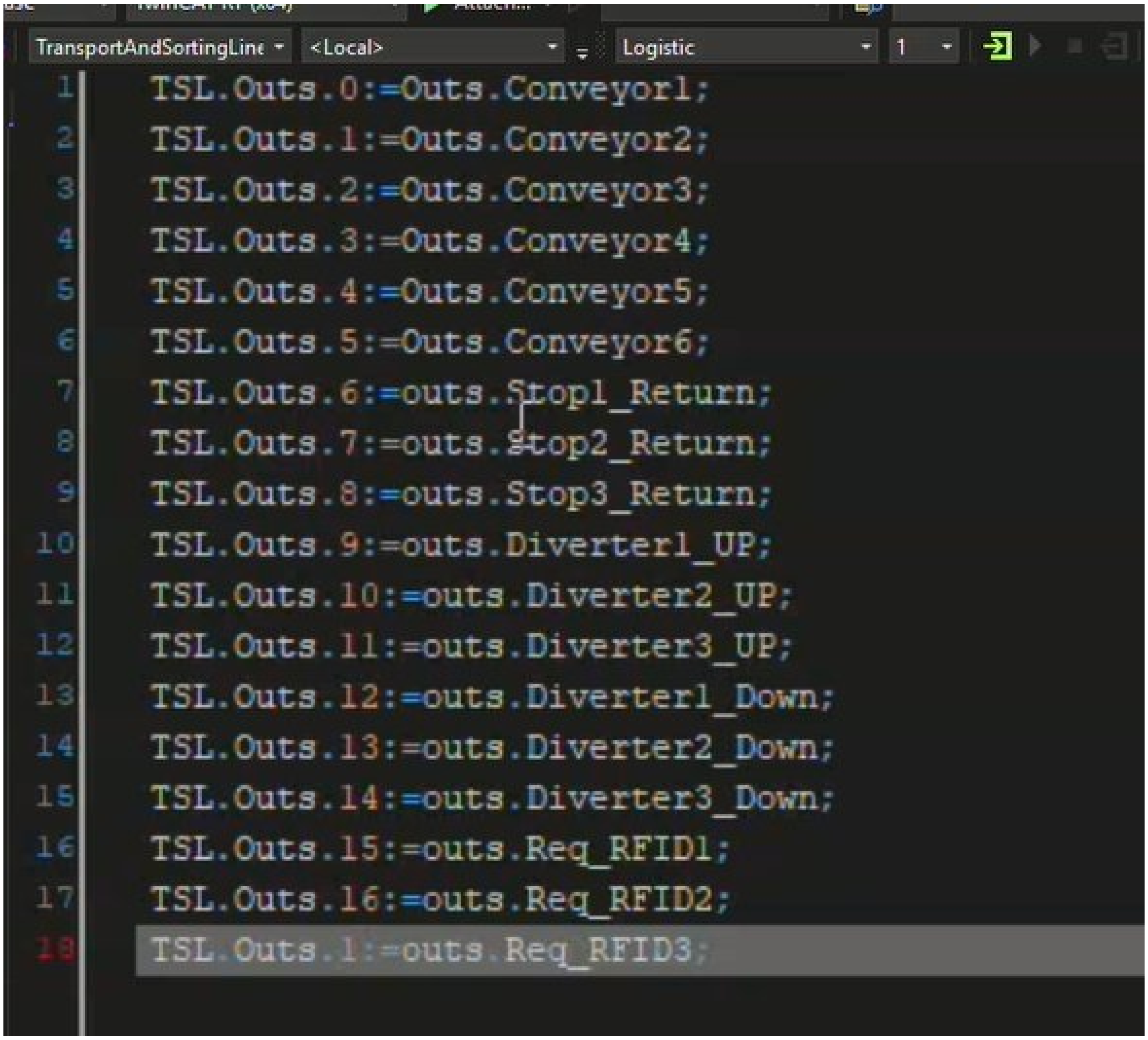


Open the action Map_Inputs, we need to map all inputs variables Please note that we can see that it will be easier to identified an Inps.Name instead of TSL.DINT.x. On SCL the assignation of a variables follow the VarName := Value;.



Please note that the screen shoot does not have all the variables. Please map all of them.

Open the action Map_Outputs, we need to map all output variables are show below.

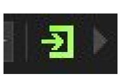



```
TSL.Outs.0:=Outs.Conveyor1;
TSL.Outs.1:=Outs.Conveyor2;
TSL.Outs.2:=Outs.Conveyor3;
TSL.Outs.3:=Outs.Conveyor4;
TSL.Outs.4:=Outs.Conveyor5;
TSL.Outs.5:=Outs.Conveyor6;
TSL.Outs.6:=outs.Stop1_Return;
TSL.Outs.7:=outs.Stop2_Return;
TSL.Outs.8:=outs.Stop3_Return;
TSL.Outs.9:=outs.Diverter1_UP;
TSL.Outs.10:=outs.Diverter2_UP;
TSL.Outs.11:=outs.Diverter3_UP;
TSL.Outs.12:=outs.Diverter1_Down;
TSL.Outs.13:=outs.Diverter2_Down;
TSL.Outs.14:=outs.Diverter3_Down;
TSL.Outs.15:=outs.Req_RFID1;
TSL.Outs.16:=outs.Req_RFID2;
TSL.Outs.1:=outs.Req_RFID3;
```

The next step is download the program and see the change of the inputs and see the reaction of the model with the outputs.

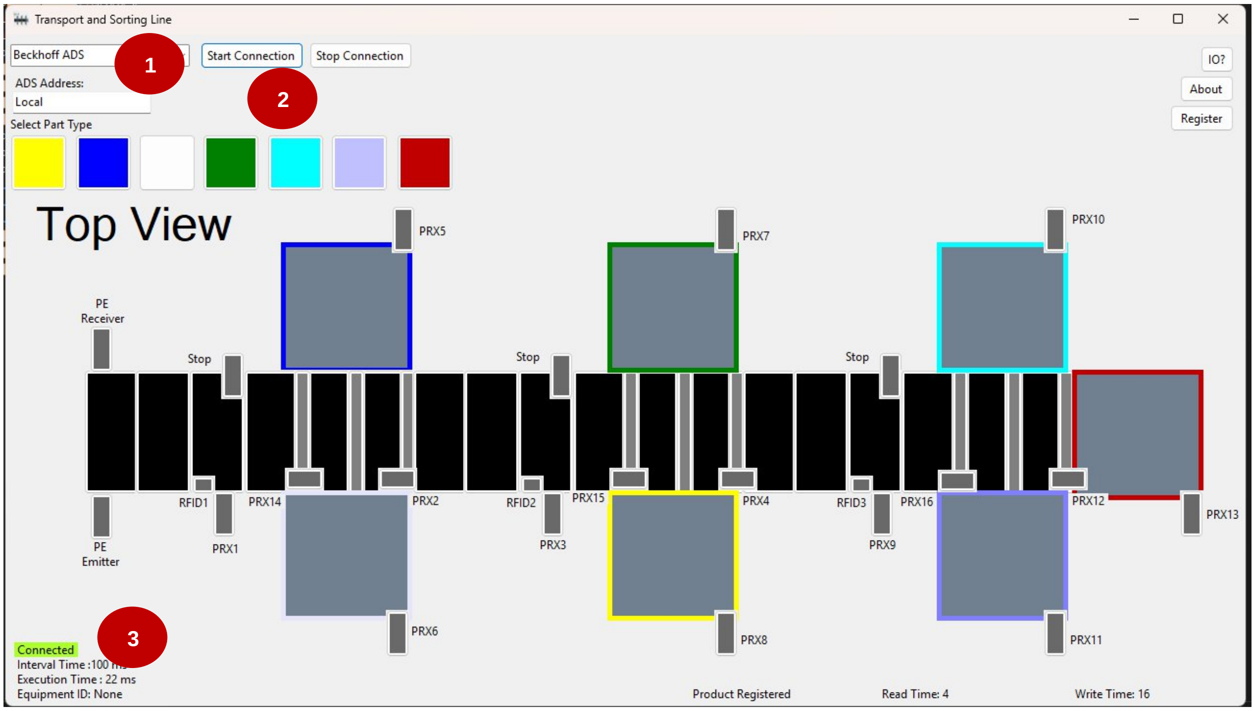
Remember to check the previous chapter, if you have questions of how to download a program.

Using the green arrow with a green frame icon is the option to download to the program to the TwinCAT Runtime.

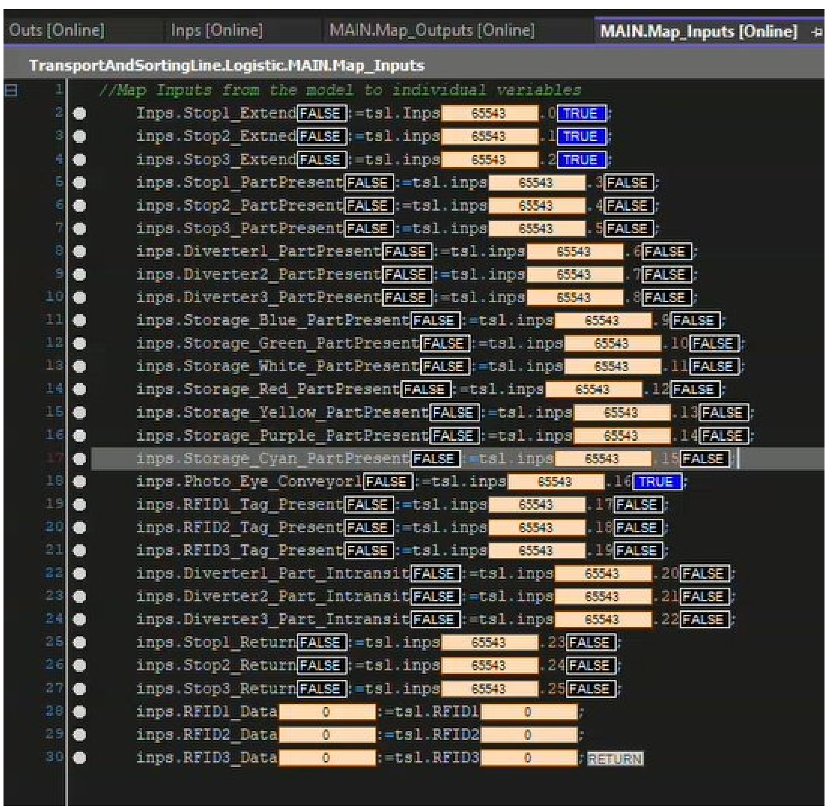


The next step is to connect the model to the TwinCAT runtime.

Select on the textbox Beckhoff ADS. At ADS address elect Local (Local connection). Then click Start Connection. We should be able to verify if the system is connect with the indicator that shows Connected on Green.



We should be able to see the reaction when we drop a box on the PE sensor located at Inps.16. In order to do that we need to open the actions Map_Inputs.



We should be able to see the reaction of the Conveyor 1 when we force the output. In order to force a variable we need to go to Map Outputs Actions.

Click on the Outs.Conveyor (it should appear False<True>). Click on the icon to download the variable changes. We should be able to see the conveyor moving the block.

Download variables.

Upload variables.

Download and upload variables.