



Knowledge Base Management

Actuator Sensor Interface Device
Communication

Contents

AS-I Communication	4
Architecture	6
Topology	7
Application Explorer	13
Project Explorer	13

Introduction

The purpose of the document is to help PES users to manage and integrate AS-I based devices into PES and harvest the splendid features of PES Library Objects. This document explains the step by step procedure of entire development cycle.

AS-I Communication

AS-Interface (AS-i) provides you with an effective and powerful bus system which connects all sensors and actuators on the lowest field level with the superior control – with unrivaled ease, safety and integration.

You can connect to an AS-Interface bus:

- Standard slaves (with addresses 1 to 31)
- Extended slaves (with addresses 1A to 31A and 1B to 31B)
- Slaves that support Combined Transaction Type profiles.

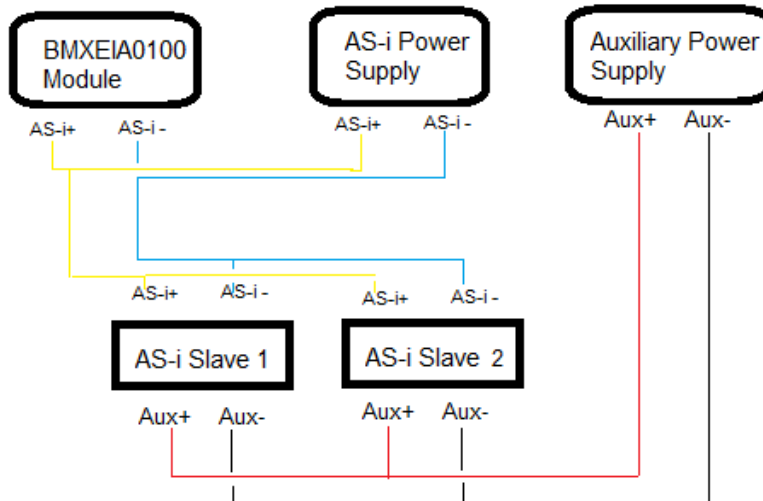
NOTE: The standard and extended slaves can be mixed on the same AS-Interface bus. Standard slaves can only be configured on bank A. If a standard slave occupies an address on bank A, an extended slave cannot be configured at the same address on bank B.

For setting the Slave Id for slave devices it is required to use the ASI configuration console .Schneider Electric proposes 'ASITERV2' console for configuration and diagnosis of AS-i modules.

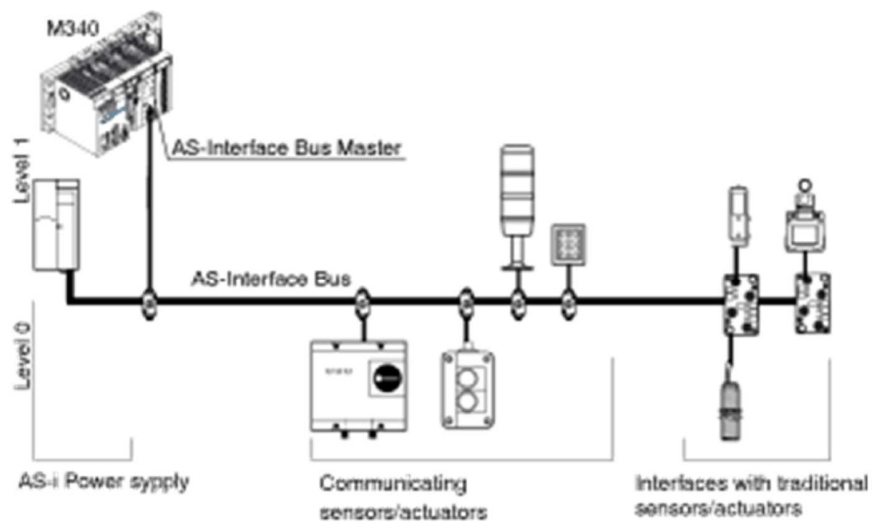


AS-i Bus Wiring

AS-i bus wiring is given below.



The illustration below is a simple example of an AS-Interface Bus:



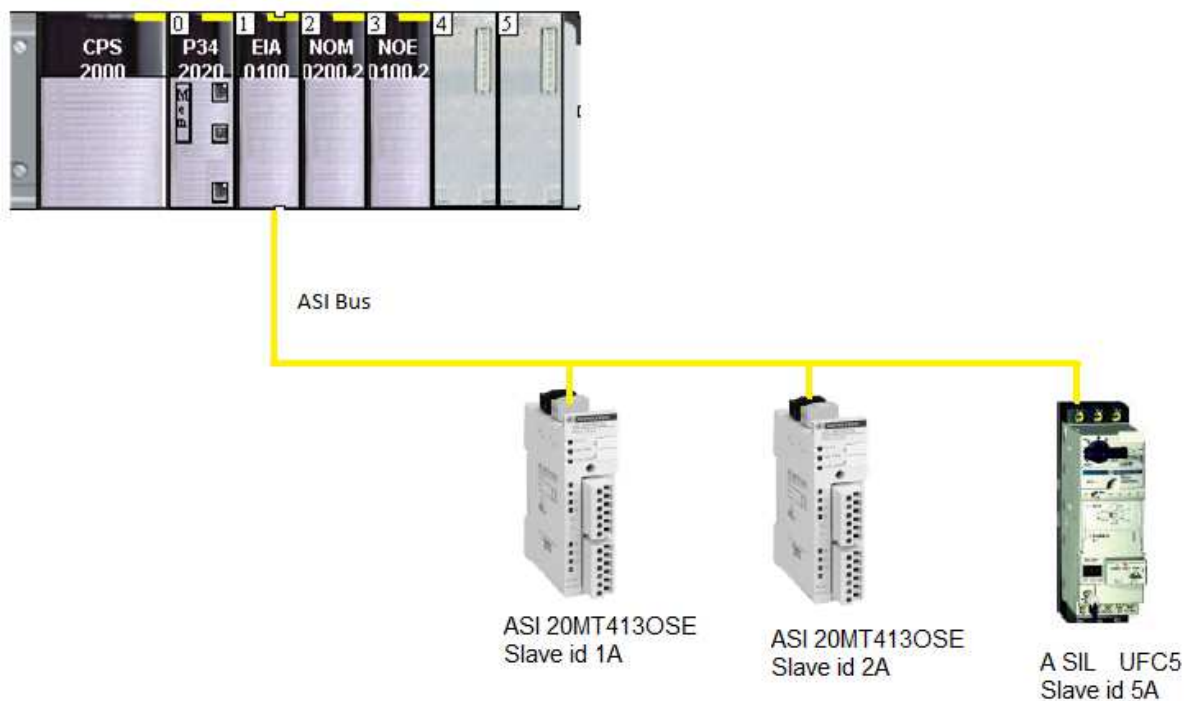
Knowledge Base Management-Actuator Sensor Interface Device Communication

In test setup we are having 3 slave devices

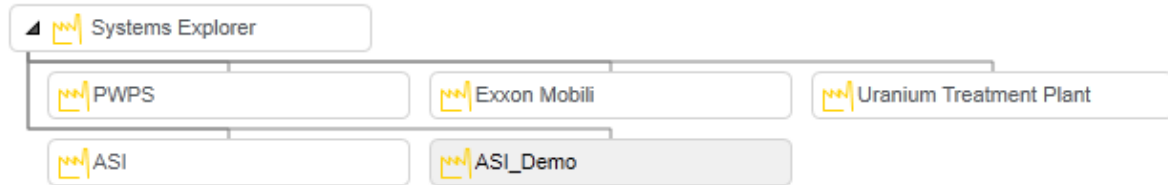
Slaved ID	Module Name	Description
1A	ASI 20MT413OSE	Digital module .4 DI, 3 DO
2A	ASI 20MT413OSE	Digital module .4 DI, 3 DO
5A	A SIL UFC5	ASI Interface module for Tesys Starter

Architecture

Architecture is based on M340 platform with BMX EIA0100 ASI interface module.



Create a System , Name it as ASI_Demo

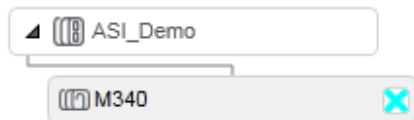


Topology

Go to Topology

Create a M340 Controller rename it as M340 for identification

Select configure option from the drop down menu obtained by right clicking the M340 controller.



Testing Rack has got following configuration

CPS2000 –Power supply module

P342020 –CPU module

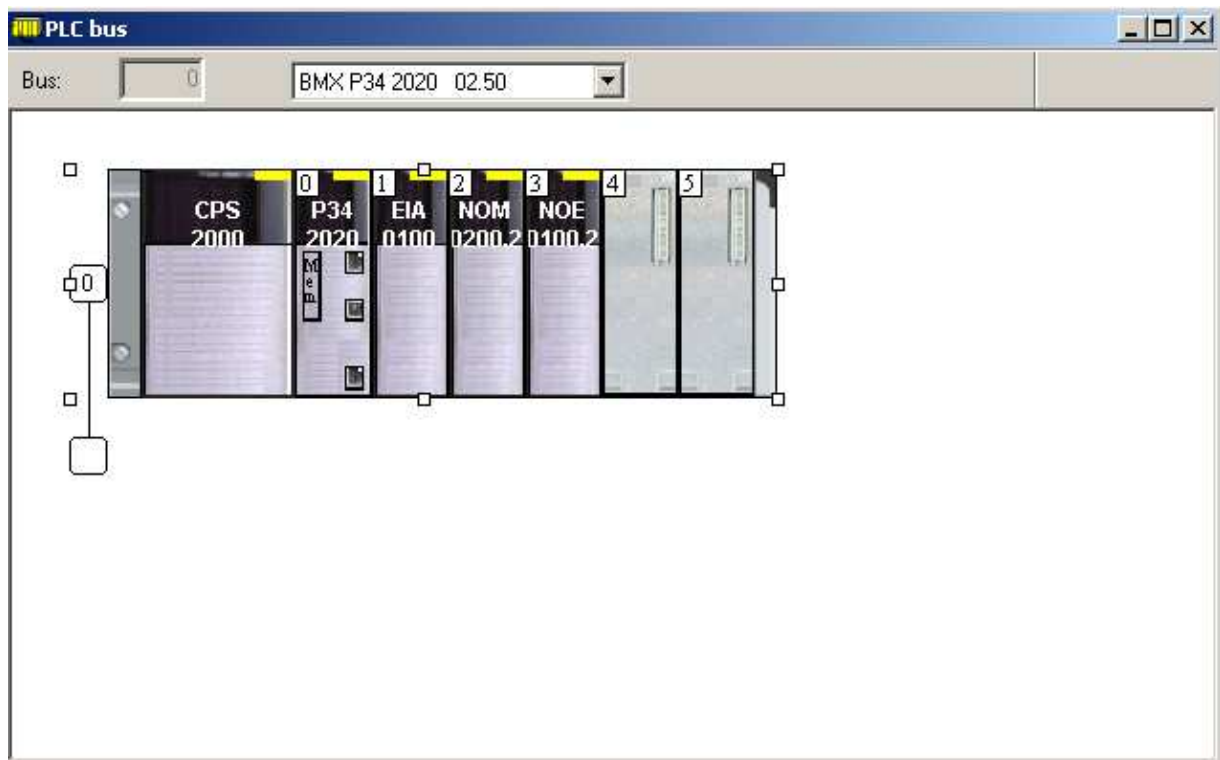
EIA0100 – AS –interface module

NOM0200- Bus module 2 RS485/232 port

NOE0100-TCP/IP module

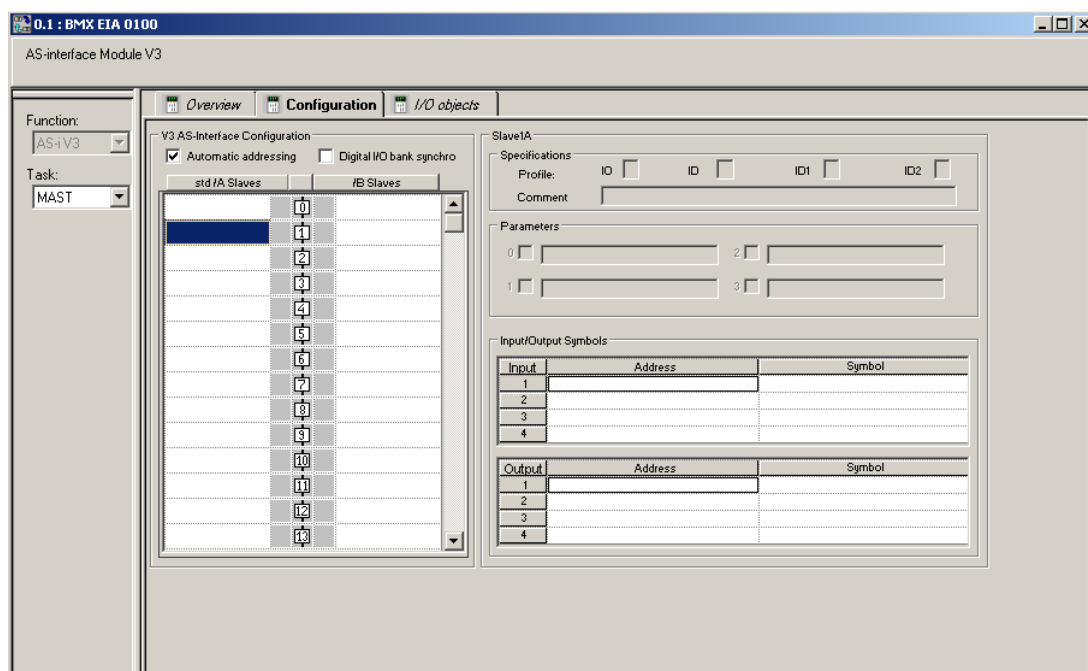
So configure accordingly

Knowledge Base Management-Actuator Sensor Interface Device Communication



Next step is to configure slave device in EIA0100 module

Double click on the EIA0100 module and go to configuration tab. Configuration tab looks like this



Knowledge Base Management-Actuator Sensor Interface Device Communication

Double click on the empty area next to slave number to open associate profile dialog box. Left group is for Group A Slaves and Right group is for Group B slaves.

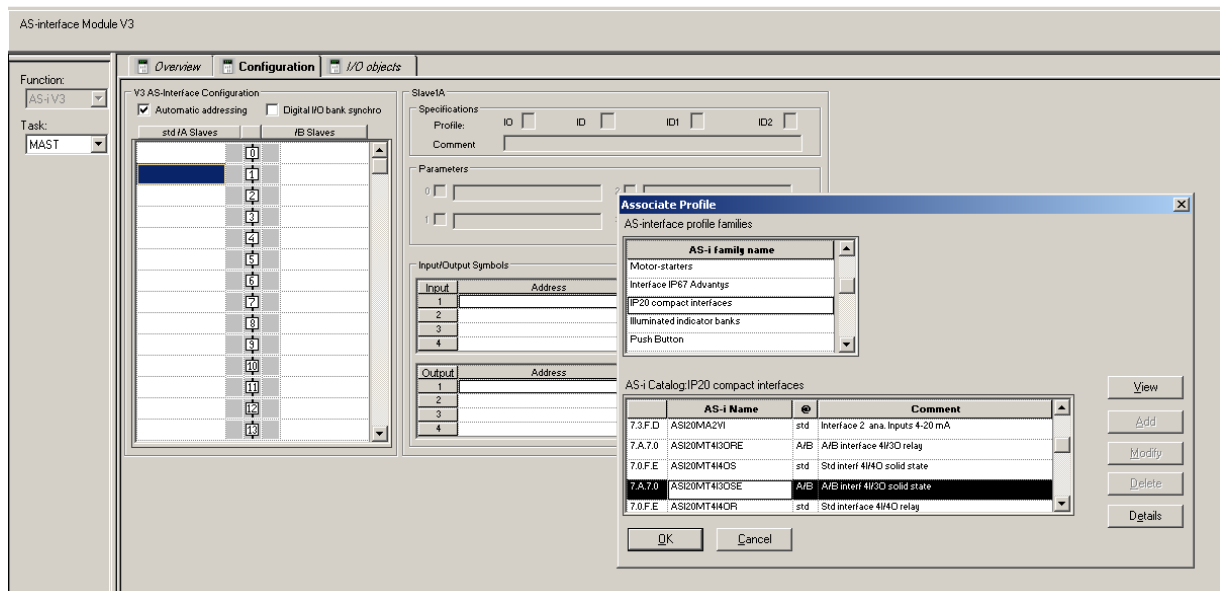
PES participant tool Unity Pro software offers a catalog of Schneider-Electric products that groups together all available AS-Interface slaves.

This catalog is currently structured according to the following families:

- Safety products
- Motor-starters
- Advantys IP67 interface
- IP20 compact interfaces
- Illuminated indicator banks
- Push button
- Out of sale
- Private family

Selecting Private family enables the user to enhance the Unity Pro catalog with specific AS-Interface products via its programming terminal.

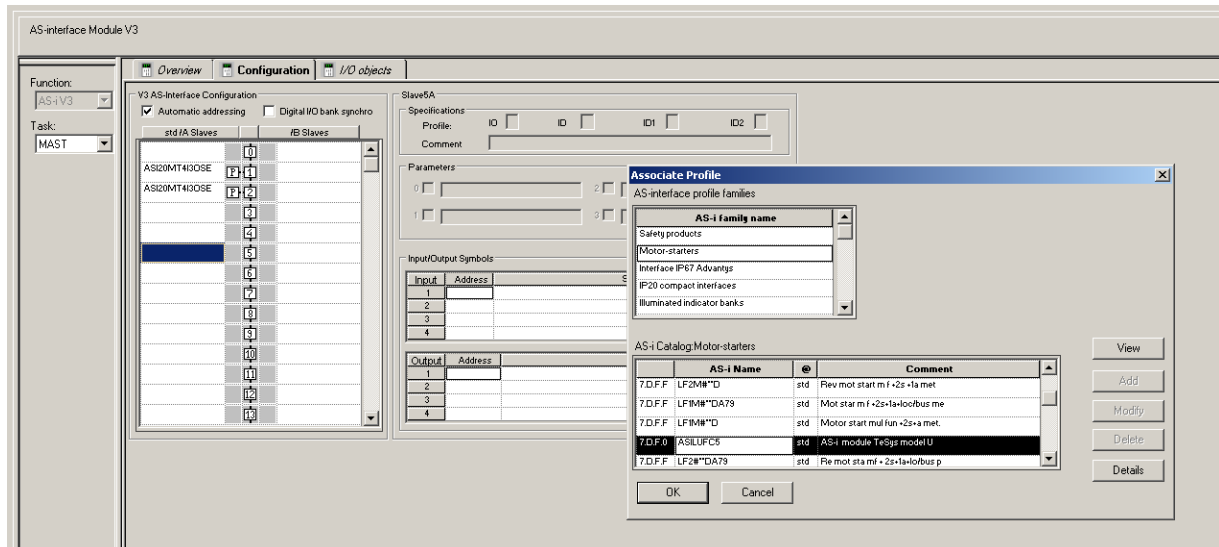
ASI 20MT413OSE module belongs to IP20 compact interfaces



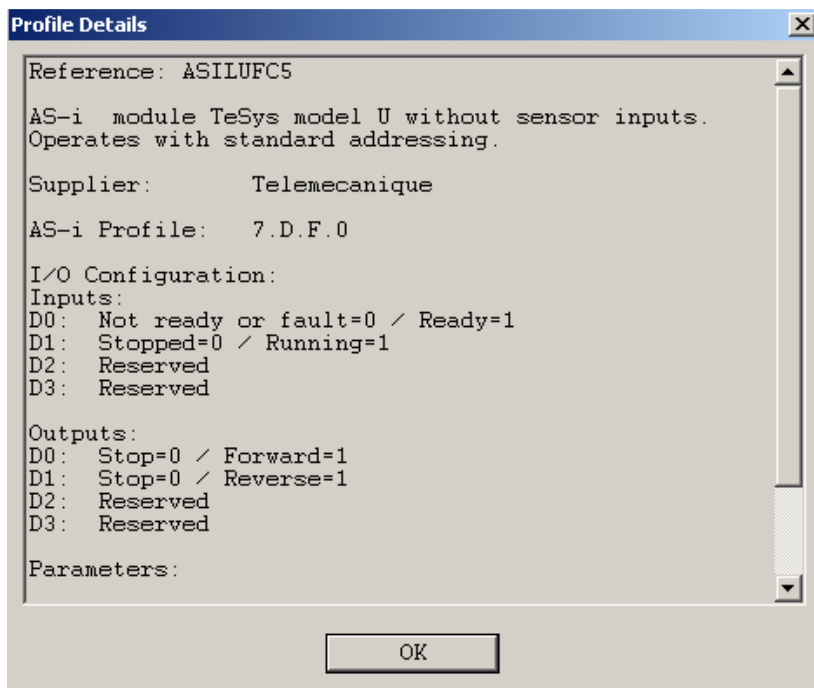
Select ASI 20MT413OSE module and click OK then ASI 20MT413OSE will get configure in 1A position. Do the same steps for 2nd ASI 20MT413OSE module by double clicking on the 2nd slot

Knowledge Base Management-Actuator Sensor Interface Device Communication

Configure A SILUFC5 device in 5th slot. A SILUFC5 belongs to motor starters family.



To see the device profile details click on the Details button.

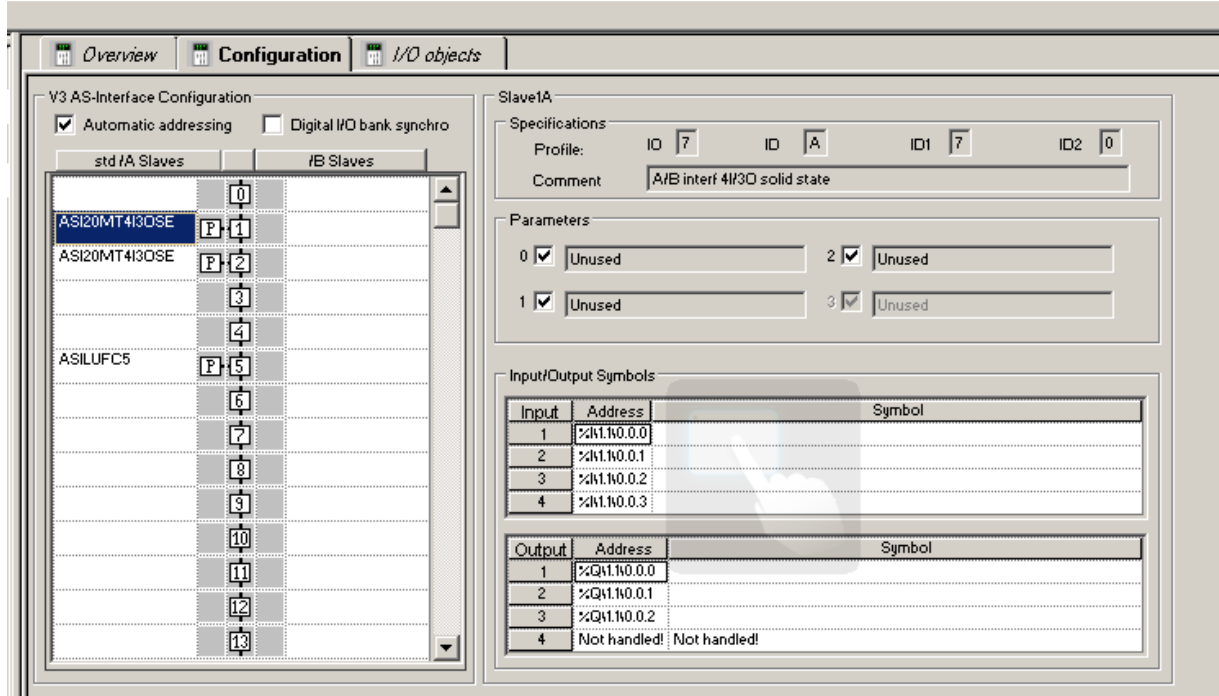


Once you configured all 3 devices in respective slave ID position .Click on the validate button on the tool bar to save the configuration.

Next step is to configure Input/Output variables of the respective device.

Knowledge Base Management-Actuator Sensor Interface Device Communication

Address details of each configured module can be seen through the table next to the device configuration box.



To create input output variables for each device go to IO objects tab.

Select %I and %Q check box to see all input and output digital variables configured in this AS Interfacing module and select update grid option.

Create variables for input and output variables of each configured ASI slaves



Examples of digital input variable addressing

%I1.1.1.0.0.0 is the address of first digital input in the device with slave id 1

%I1.1.1.0.1.0 is the address of second digital input in the device with slave id 1

%I1.2.1.0.0.0 is the address of first digital input in the device with slave id 2

Knowledge Base Management-Actuator Sensor Interface Device Communication

%I\1.5\1.0.0.0 is the address of first digital input in the device with slave id 5

Examples of digital output variable addressing

%Q\1.1\1.0.0.0 is the address of first digital output in the device with slave id 1

%Q\1.1\1.0.1.0 is the address of second digital output in the device with slave id 1

%Q\1.2\1.0.1.0 is the address of second digital output in the device with slave id 2

%Q\1.5\1.0.1.0 is the address of second digital output in the device with slave id 5

Note: It is essential to note down the variables created here because the variables created in topology will not be available in the project explorer. While using these variables in project explorer variable not declared error will come but it can be ignored. This error will not reflect in the executable build.

Here all the Input and output variables associated with all the ASI slave devices are created.

The screenshot shows the 'I/O objects' configuration window. On the left, there are sections for 'I/O variable creation' and 'I/O Objects'. The 'I/O variable creation' section has fields for 'Prefix for name:', 'Type:' (set to 'EBOOL'), and 'Comment:', with a 'Create' button. The 'I/O Objects' section has checkboxes for 'Channel:' (with '%CH' selected), 'Configuration:' (with '%KW', '%KD', and '%KF' selected), 'System:' (with '%MW' selected), 'Status:' (with '%MW' selected), 'Parameter:' (with '%MW', '%MD', and '%MF' selected), 'Command:' (with '%MW', '%MD', and '%MF' selected), and 'Implicits:' (with '%I', '%W', '%D', '%F', and '%I ERR' selected). There are 'Update grid' and 'Filter on usage' buttons at the bottom of this section.

	Address	Name	Type	Comment
1	%I0.1.0		EBOOL	
2	%I0.1.0.1		EBOOL	
3	%I\1.1\0.0.0.0	ASI_S1_IP_00	EBOOL	ASI_Slave1_Input_0
4	%I\1.1\0.0.1.0	ASI_S1_IP_01	EBOOL	ASI_Slave1_Input_1
5	%I\1.1\0.0.2.0	ASI_S1_IP_02	EBOOL	ASI_Slave1_Input_2
6	%I\1.1\0.0.3.0	ASI_S1_IP_03	EBOOL	ASI_Slave1_Input_3
7	%Q\1.1\0.0.0.0	ASI_S1_OP_00	EBOOL	ASI_Slave1_Output_0
8	%Q\1.1\0.0.1.0	ASI_S1_OP_01	EBOOL	ASI_Slave1_Output_1
9	%Q\1.1\0.0.2.0	ASI_S1_OP_02	EBOOL	ASI_Slave1_Output_2
10	%Q\1.1\0.0.3.0	ASI_S1_OP_03	EBOOL	ASI_Slave1_Output_4
11	%I\1.2\0.0.0.0	ASI_S2_IP_00	EBOOL	ASI_Slave2_Input_0
12	%I\1.2\0.0.1.0	ASI_S2_IP_01	EBOOL	ASI_Slave2_Input_1
13	%I\1.2\0.0.2.0	ASI_S2_IP_02	EBOOL	ASI_Slave2_Input_2
14	%I\1.2\0.0.3.0	ASI_S2_IP_03	EBOOL	ASI_Slave2_Input_3
15	%Q\1.2\0.0.0.0	ASI_S2_OP_00	EBOOL	ASI_Slave2_Output_0
16	%Q\1.2\0.0.1.0	ASI_S2_OP_01	EBOOL	ASI_Slave2_Output_1
17	%Q\1.2\0.0.2.0	ASI_S2_OP_02	EBOOL	ASI_Slave2_Output_2
18	%Q\1.2\0.0.3.0	ASI_S2_OP_03	EBOOL	ASI_Slave2_Output_3
19	%I\1.5\0.0.0.0	ASI_S5_IP_00	EBOOL	ASI_Slave5_Input_0
20	%I\1.5\0.0.1.0	ASI_S5_IP_01	EBOOL	ASI_Slave5_Input_1
21	%I\1.5\0.0.2.0	ASI_S5_IP_02	EBOOL	ASI_Slave5_Input_2
22	%I\1.5\0.0.3.0	ASI_S5_IP_03	EBOOL	ASI_Slave5_Input_3
23	%Q\1.5\0.0.0.0	ASI_S5_OP_00	EBOOL	ASI_Slave5_Output_0
24	%Q\1.5\0.0.1.0	ASI_S5_OP_01	EBOOL	ASI_Slave5_Output_1
25	%Q\1.5\0.0.2.0	ASI_S5_OP_02	EBOOL	ASI_Slave5_Output_2
26	%Q\1.5\0.0.3.0	ASI_S5_OP_03	EBOOL	ASI_Slave5_Output_3

Keep these variables for further usage in project explorer in refinement.

Knowledge Base Management-Actuator Sensor Interface Device Communication

Configurations required for EIA0100 and slave devices are completed.

NOE and NOM module communication settings needs to done separately that are not shown here.

Save the configuration.

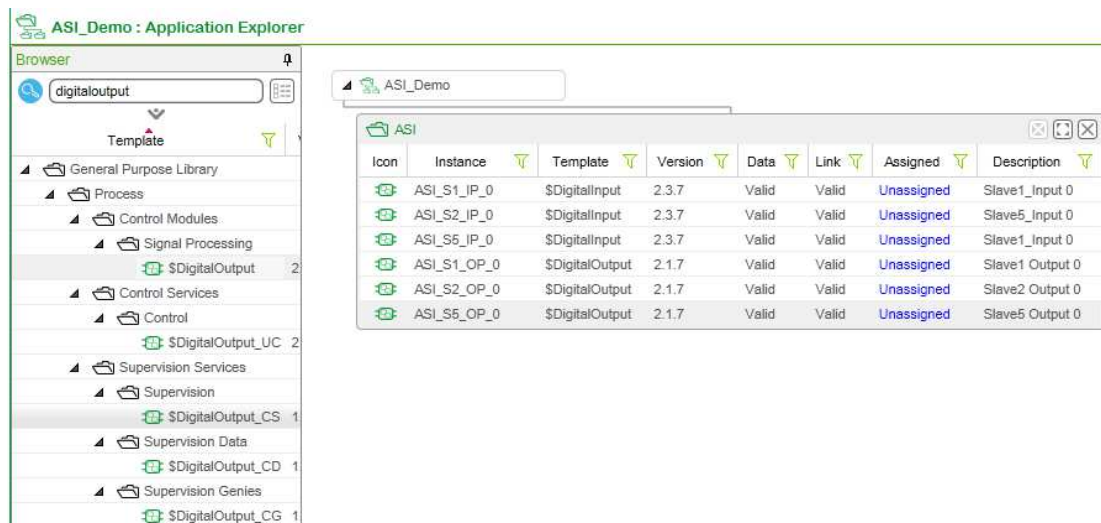
In PES Topology explorer create an Ethernet network

Link NOE and CPU network to the Ethernet network created.

Application Explorer

Now go to Application Explorer

In application Explorer create instances for digital input and digital output. For test purpose only instance for 1 digital input and digital output from each slave device shall be created.



Project Explorer

Next step is to assign the instantiated block.

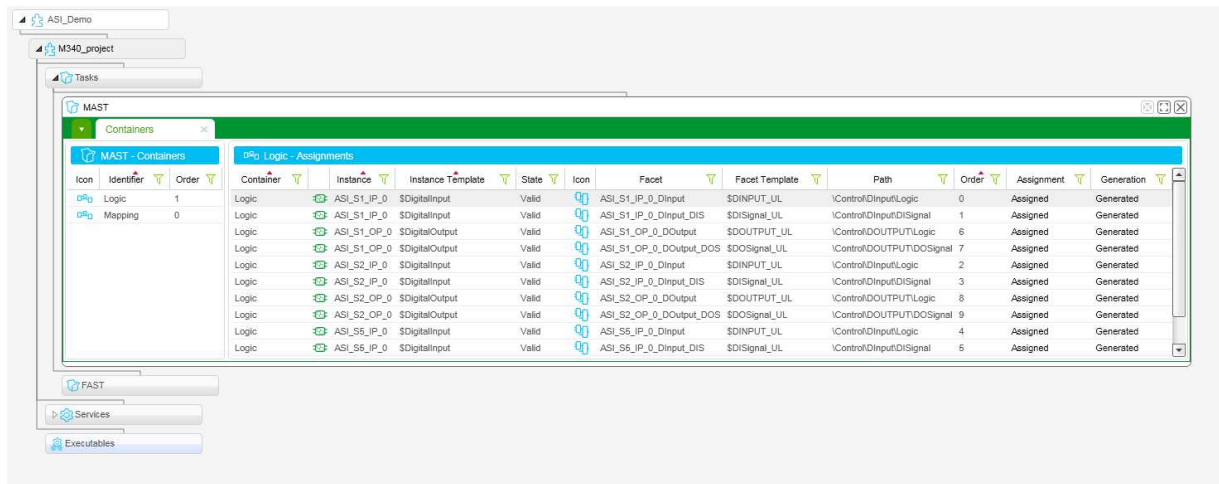
Go to Project Explorer

Create a control project with M340 controller and rename it as 340_project

Create 1 MAST section

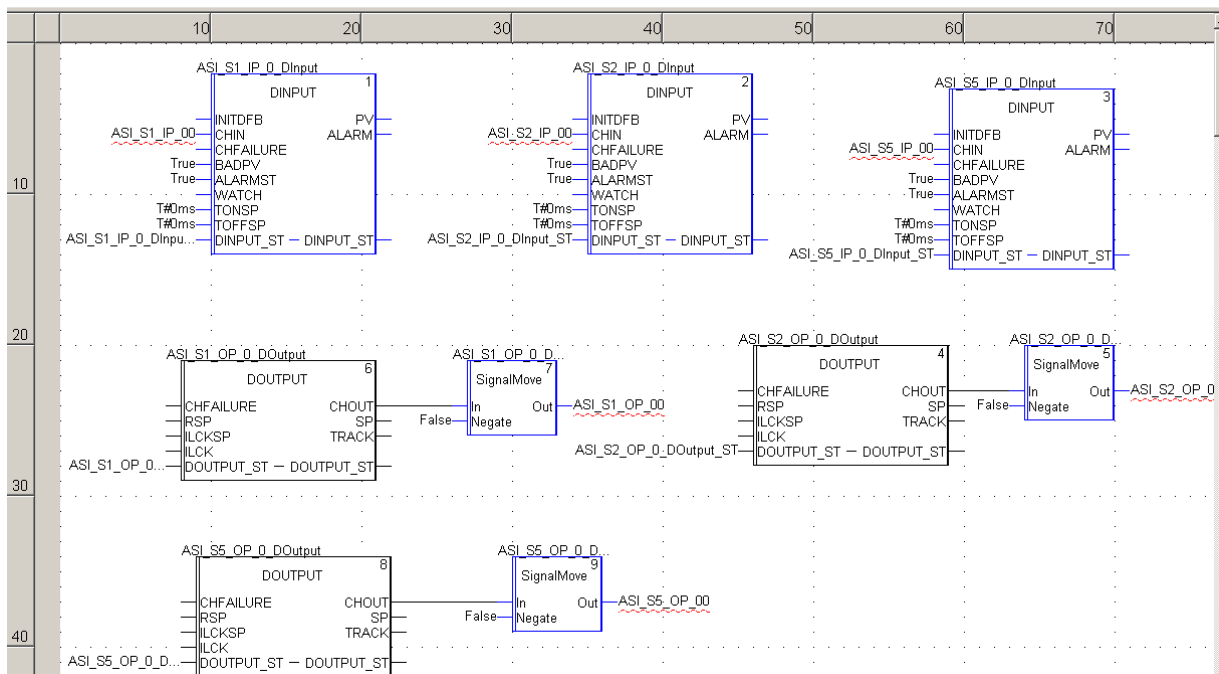
Knowledge Base Management-Actuator Sensor Interface Device Communication

Assign Facets to the Logic section and generate the project



Refine the project for mapping the variables created in the topology to the channel input of the 'digitalinput' and 'digitaloutput' instances

Input ASI variables are connected to the channel input pin of 'Digitalinput' block instantiated in the section and channel output variable of 'DigitalOutput' is connected to Output ASI variables which are created in topological explorer.



Variables created in the topology are represented as undefined variables and its underlined in red color.

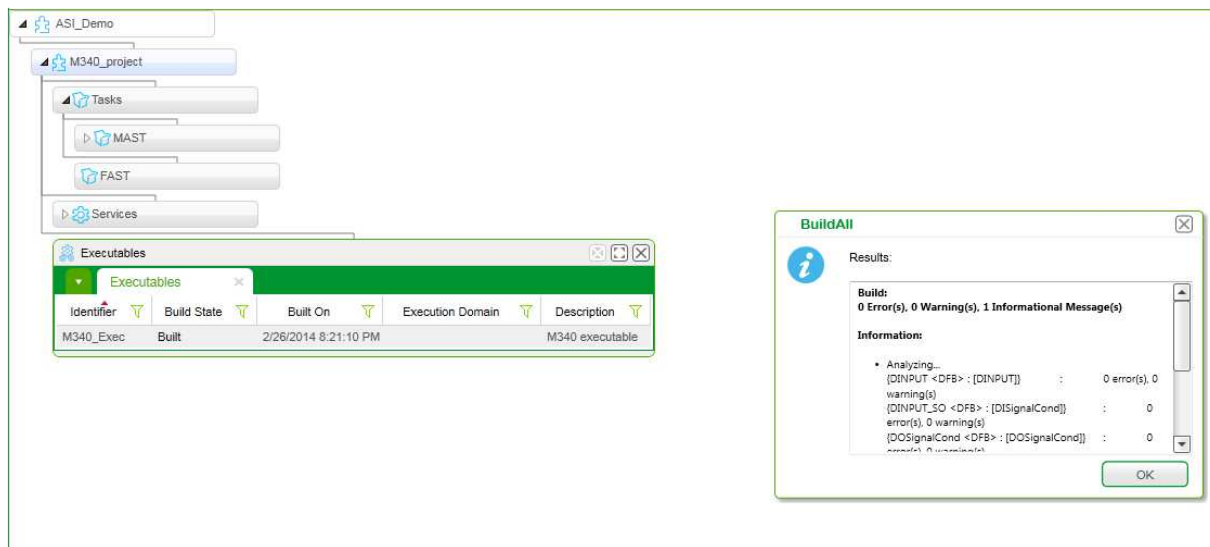
Knowledge Base Management-Actuator Sensor Interface Device Communication

Save the changes made in refinement and generate the project.

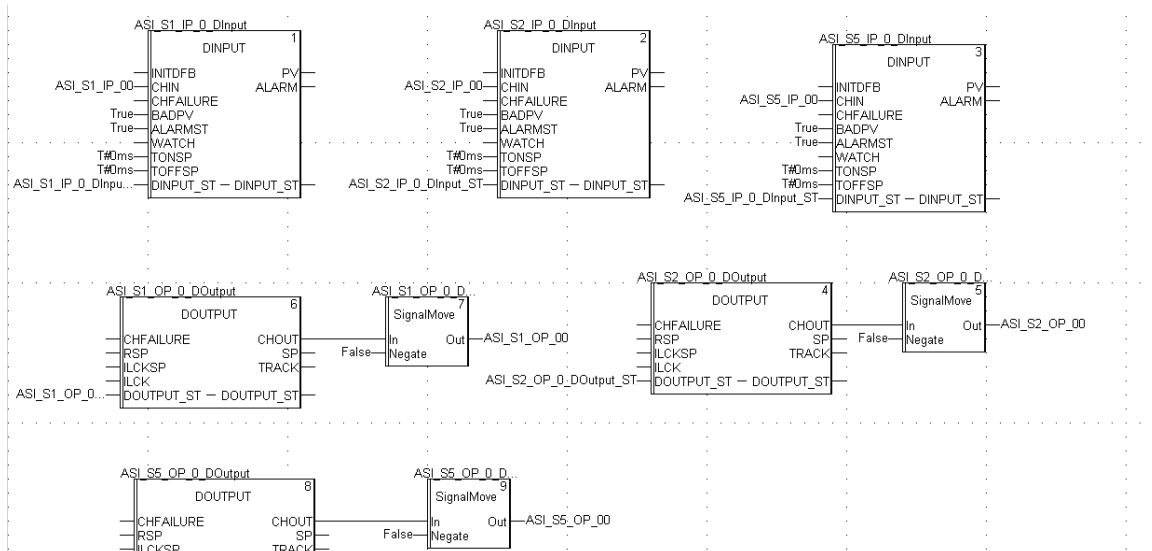
Next step is to create the executables, rename it as M340_Exec and map service to the M340 controller created in the Topology Explorer.

Build the project.

Once the project build is completed make sure that no error is shown in the build result



Open the built project and make sure that changes made during refinement are reflected as expected.



Download the project into the controller and Confirm that ASI inputs and outputs are updating exactly as expected.