Chapter 6 Download

TABLE OF CONTENTS

6.1 General Information

6.1.1 Objectives

Upon completion of this chapter, you will be able to:

- Set up communication to controller AC 800M
- Download and test hardware in online mode

6.1.2 Legend

>	Indicates when you go from one menu to a sub-menu
Italic	Indicates object and file names
“ “	Indicates dialog box buttons, tabs, menus etc.
Bold	Indicates important topics
☞	Indicates start/explanation of student activity

6.1.3 Reference Documentation

3BSE036351	Industrial IT 800xA – Control and I/O AC 800M Controller - Hardware and Operation
3BSE041548	Industrial IT - Compact Control Builder AC 800M Getting Started – Introduction and Installation
3BSE040935	Industrial IT – Compact Control Builder AC 800M Basic Control Software – Introduction and Configuration

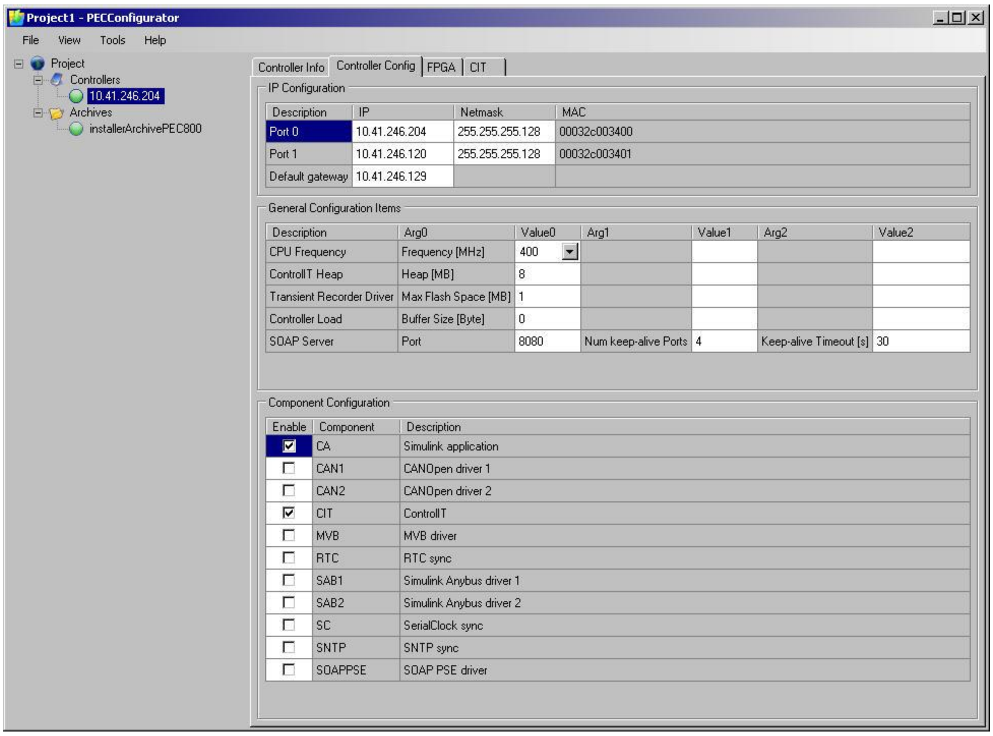
6.2 Communication to AC800M Controller

All controllers delivered from the factory have the same IP address. You must change the IP address to an address, which is available in your own network.

6.2.1 How to Configure the IP Address with AC 800PEC

A different procedure needs to be applied if an AC 800PEC controller is used. In order to manage the configuration of this controller (i.e. also configure the TCP/IP setting) a tool called PEC Configurator is used.

The PEC Configurator reads the configuration from the controller and displays it to the user. The user has the possibility to change the configuration of the controller and store the modified configurations back to the target.



IP Configuration

Description	IP	Netmask	MAC
Port 0	10.41.246.204	255.255.255.128	00032c003400
Port 1	10.41.246.120	255.255.255.128	00032c003401
Default gateway	10.41.246.129		

General Configuration Items

Description	Arg0	Value0	Arg1	Value1	Arg2	Value2
CPU Frequency	Frequency [MHz]	400				
ControlIT Heap	Heap [MB]	8				
Transient Recorder Driver	Max Flash Space [MB]	1				
Controller Load	Buffer Size [Byte]	0				
SOAP Server	Port	8080	Num keep-alive Ports	4	Keep-alive Timeout [s]	30

Component Configuration

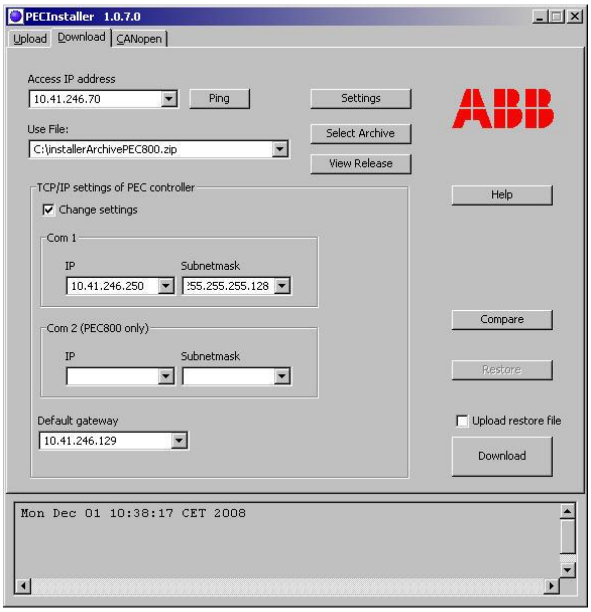
Enable	Component	Description
☑	CA	Simulink application
□	CAN1	CANOpen driver 1
□	CAN2	CANOpen driver 2
☑	CIT	ControlIT
□	MVB	MVB driver
□	RTC	RTC sync
□	SAB1	Simulink Anybus driver 1
□	SAB2	Simulink Anybus driver 2
□	SC	SerialClock sync
□	SNTP	SNTP sync
□	SOAPPSE	SOAP PSE driver

Please, refer to the AC 800PEC PEC Configurator User Manual for a detailed description of this tool.

You might also use the PC Installer for changing the IP-Address

6.2.2 How to Download the Firmware with AC 800PEC

A tool called PEC Installer is used either to update the platform- and control software of an AC 800PEC controller and to backup the software from an AC 800PEC controller to the PC: in the PEC Installer, the first use case is called 'Download', the second one is called ‘Upload’.



PECInstaller 1.0.7.0

Access IP address: 10.41.246.70

Use File: C:\installerArchivePEC800.zip

TCP/IP settings of PEC controller

☑ Change settings

Com 1

IP: 10.41.246.250

Subnetmask: :55.255.255.128

Com 2 (PEC800 only)

IP:

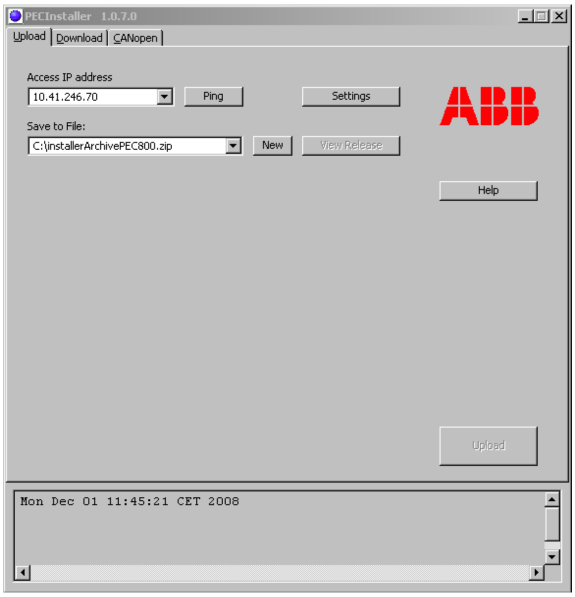
Subnetmask:

Default gateway: 10.41.246.129

□ Upload restore file

```
Mon Dec 01 10:38:17 CET 2008
```

Do a backup (upload):



PECInstaller 1.0.7.0

Access IP address: 10.41.246.70

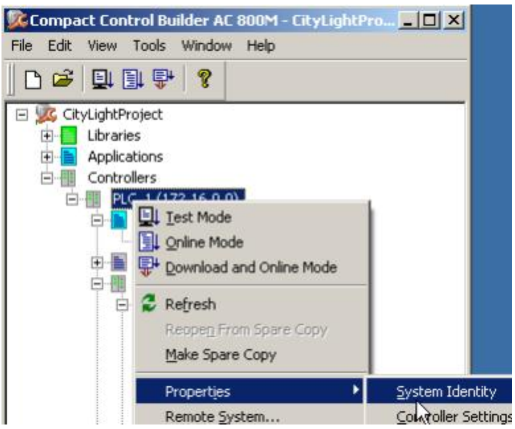
Save to File: C:\installerArchivePEC800.zip

```
Mon Dec 01 11:45:21 CET 2008
```

6.2.3 How to Configure the IP Address in the CBM

After setting the controller's IP address, you must do the IP address settings of the controller in CBM in two steps:

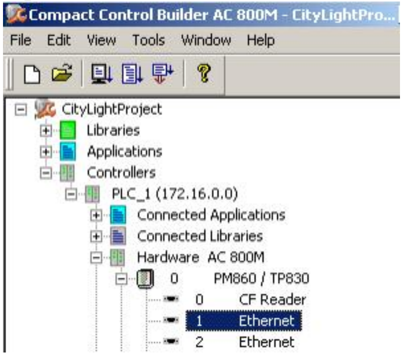
1. Right click the *Controller,* choose "Properties > System Identity" and then set the desired IP address of the controller.



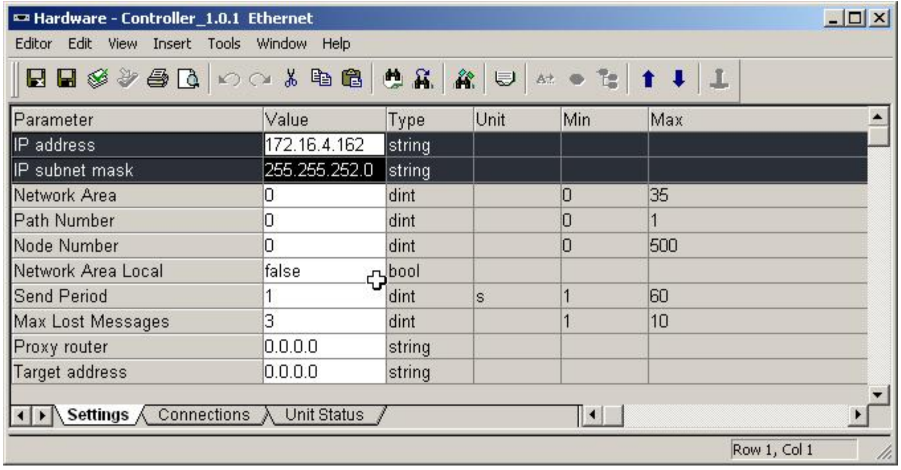
The System Identity is the network address of the selected controller. The network address is shown beside the controller icon in the Project Explorer tree.

NOTE! The default address is 172.16.0.0

2. Open the controller's *PM8xx/TP830* and then double click the first Ethernet.



This will open a new window, click on the "Settings" tab, then set the same IP address and subnet mask as you have set with the IPConfig tool.


Hardware - Controller_1.0.1 Ethernet
Editor Edit View Insert Tools Window Help
Settings
Connections
Unit Status
Row 1, Col 1

Parameter	Value	Type	Unit	Min	Max
IP address	172.16.4.162	string			
IP subnet mask	255.255.252.0	string			
Network Area	0	dint		0	35
Path Number	0	dint		0	1
Node Number	0	dint		0	500
Network Area Local	false	bool			
Send Period	1	dint	s	1	60
Max Lost Messages	3	dint		1	10
Proxy router	0.0.0.0	string			
Target address	0.0.0.0	string			

NOTE! **The system identity should normally correspond to the setting for Ethernet 1 on the AC800M.**

6.3 Application Download

6.3.1 General

During a download, the code that has been written is checked and compiled. The controller receives the compiled application and begins executing it immediately.

Once your application has been compiled, you can:

- **Download the application and go Online.**
  This updates the application in the controller.
- **Go online without download of the application.**
  Useful when, for example, monitoring values in the controller without disturbing the controller.

A version analysis of the applications is made when you download. The analysis results are used to determine if and how applications shall be updated to new versions in the controllers.

For the AC800PEC it is always recommended to load the application to the

because of different save cold retain value handling

6.3.2 Restart Options

When the system restarts, variables with No Retain, Retain and Cold Retain attributes will behave differently. It will also matter if the system performs a warm or a cold restart.

WARM RESTART at AC800M only

At a warm restart, variables with the attribute <retain> or <coldretain> are protected by the battery during restart. During the time that the controller is stopped, all output I/Os will keep their last value.

COLD RESTART

At a cold restart, variables with the attribute <coldretain> keep their values, since they are stored on the hard disk. Variables without the attribute <coldretain> lose their values. During the stop time, all output I/Os will keep their last safe state/OSP values.

INIT RESTART

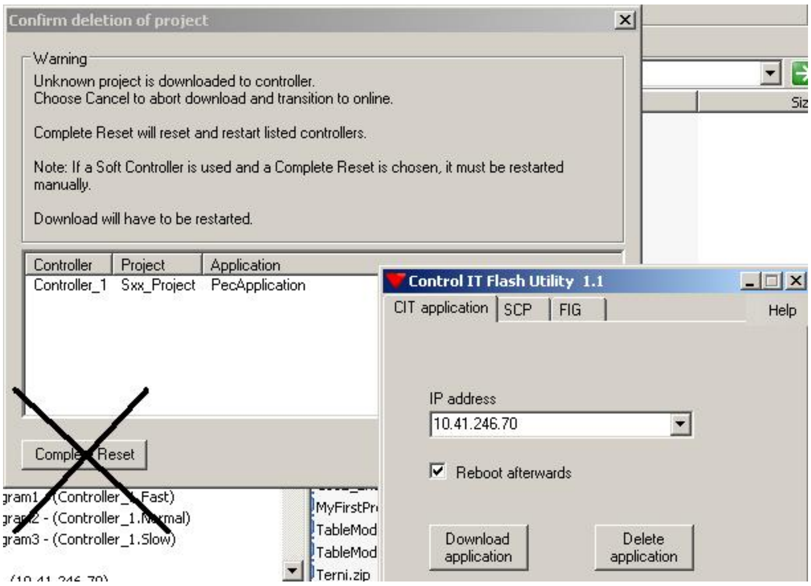
At a Init restart, variables with the attribute <coldretain> or <retain> are returned to the initial value

6.3.3 Download to Flash

For the AC800PEC use the Flash Utility tool to download. But first right click on the controller and “make” the compact flash.

Select “Programs > ABB Industrial IT AC 800 PEC….”

At the very first time when you download to Ram only following windows will pop up:



Confirm deletion of project

Warning

Unknown project is downloaded to controller.
Choose Cancel to abort download and transition to online.

Complete Reset will reset and restart listed controllers.

Note: If a Soft Controller is used and a Complete Reset is chosen, it must be restarted manually.

Download will have to be restarted.

Controller	Project	Application
Controller_1	Sxx_Project	PecApplication

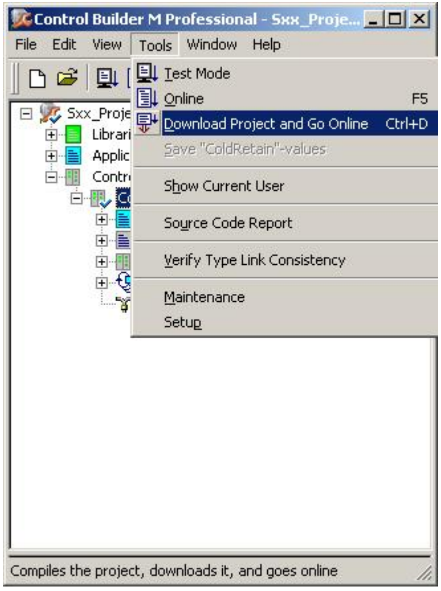
Control IT Flash Utility 1.1

IP address: 10.41.246.70

☑ Reboot afterwards

6.3.4 Download to RAM only

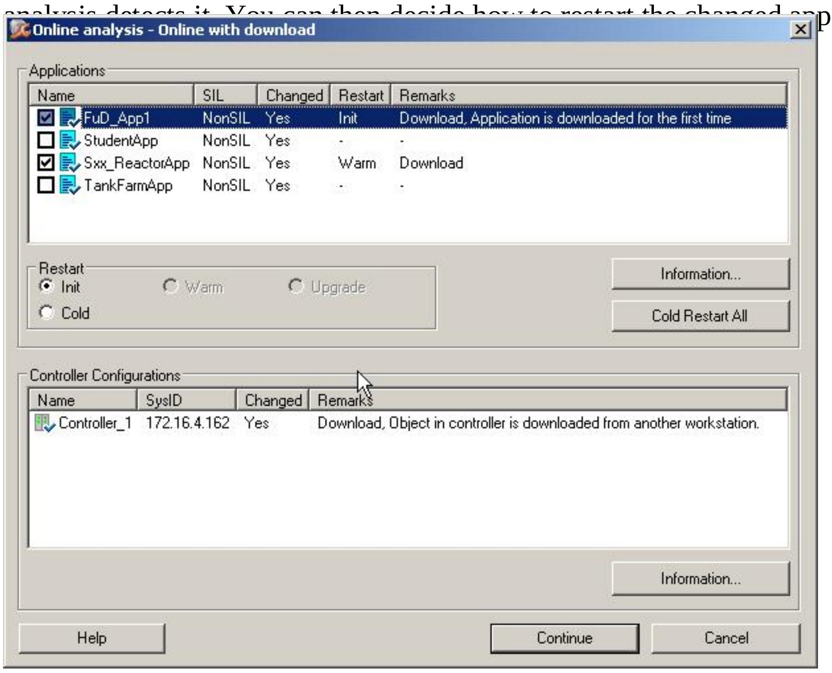
Select “Tools > Download Project and go Online” in the CBM.



Control Builder M Professional - Sxx_Proje...

Compiles the project, downloads it, and goes online

If changes have been made to one or several applications in Offline mode, the version analysis detects it. You can then decide how to restart the changed applications.



Online analysis - Online with download

Applications

	Name	SIL	Changed	Restart	Remarks
☑	FuD_App1	NonSIL	Yes	Init	Download, Application is downloaded for the first time
□	StudentApp	NonSIL	Yes	-	-
☑	Sxx_ReactorApp	NonSIL	Yes	Warm	Download
□	TankFarmApp	NonSIL	Yes	-	-

Restart: ● Init ○ Warm ○ Upgrade ○ Cold

Controller Configurations

Name	SysID	Changed	Remarks
Controller_1	172.16.4.162	Yes	Download, Object in controller is downloaded from another workstation.

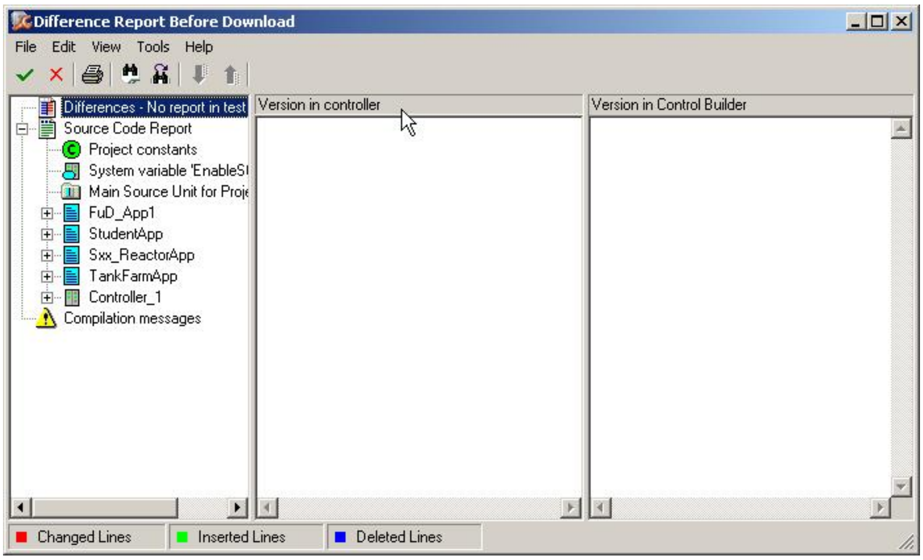
Independent of how you restart the applications, the changed application parts are downloaded to the controllers. The controller will stop the applications and restart with the new and changed application.

NOTE! **Following download the last message should be:**
You are now Online!

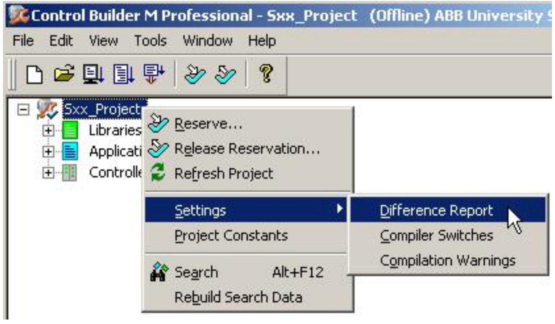
6.3.5 Difference Report

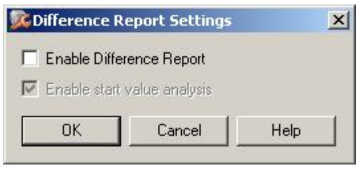
Just before the download to controllers, the system will open a ‘Difference Report’ window, which shows the differences between the applications on disk & applications in the controllers. At this time you may decide to continue or to cancel the download.

The purpose of the difference report is validation of configuration changes.



You may disable the difference report function in project context menu “Settings > Difference Report”.





☐ Enable Difference Report

☑ Enable start value analysis

6.4 Online Functions

6.4.1 General

In Online mode the Control Builder is connected to one or more controllers and offers many diagnostic and inspection features when online to the controller. You can:

- View the executing code dynamically
- Force inputs and outputs at the I/O modules
- Rescale analogue input and output channels
- Change values of variables
- Inspect values of variables

To enter Online mode, go to the tool bar menu of the Control Builder, click on "Tools" and then click on "Online".

6.4.2 Status of I/O-Signals

Use the status tab of the hardware configuration editor, in online mode, for dynamic online display of I/O channel values and forcing. I/O channel values are updated once every second.

NOTE! The tab is available in Offline mode, but is not editable.

Underflow, overflow, channel and unit errors in online mode are indicated in the Channel value cell by the following icons:

Signal Errors

↓ underflow	The signal is below the range.
↑ overflow	The signal is above the range.

Hardware Errors

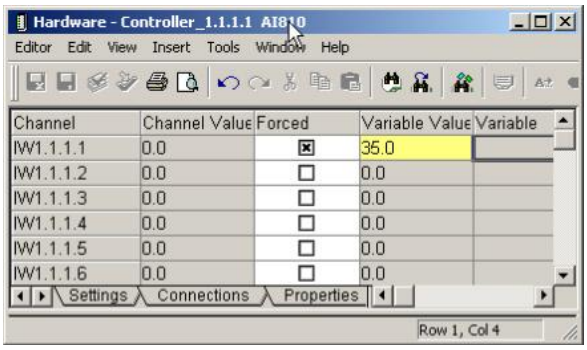
▲channel error	The channel can have under range, power-up or field power errors.
▲Unit error	The unit can be missing or of the wrong type.

6.4.3 Forcing

Forcing of I/O is possible in online mode. Check a check box in the **Forced** column and then type in a value in the **Channel** value column for outputs and in the **Variable** value column for inputs. This value overrides the values in each of those columns.

When a channel is forced all copying between the I/O value and the application value is stopped. The forced value is different for inputs and outputs.

- For inputs the forcing changes the variable value to the application.
- For outputs the forcing changes the physical I/O channel value.

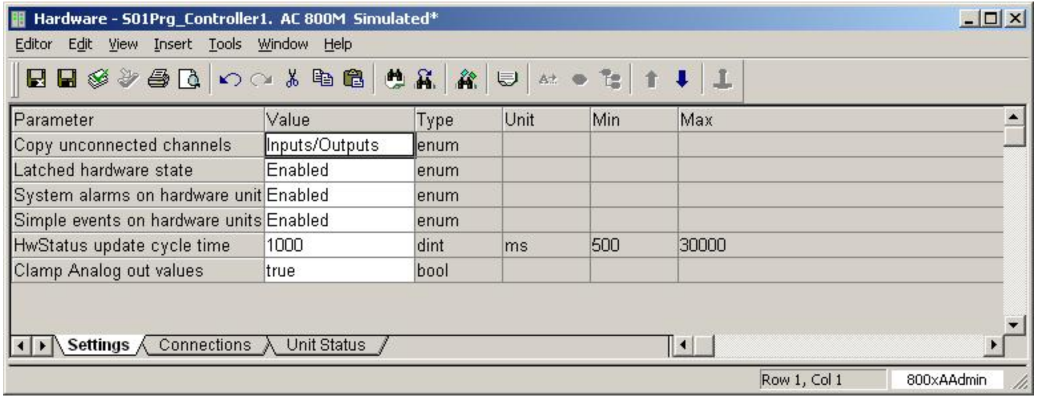

Hardware - Controller_1.1.1.1 AI810
Editor Edit View Insert Tools Window Help
Settings
Connections
Properties
Row 1, Col 4

Channel	Channel Value	Forced	Variable Value	Variable
IW1.1.1.1	0.0	☒	35.0	
IW1.1.1.2	0.0	☐	0.0	
IW1.1.1.3	0.0	☐	0.0	
IW1.1.1.4	0.0	☐	0.0	
IW1.1.1.5	0.0	☐	0.0	
IW1.1.1.6	0.0	☐	0.0	

NOTE! The background of the forced cell changes to yellow to indicate forcing.

Forcing is performed in the hardware configuration editor under the “Status” tab or the POU editor.

Normally only channels with variable connections to application programs can be forced. However, if no variable is connected you have to change the parameter **Copy Unconnected Channels** under the “Settings” tab for the current controller to get a status update.


Hardware - 501Prg_Controller1. AC 800M Simulated*
Editor Edit View Insert Tools Window Help
Settings
Connections
Unit Status
Row 1, Col 1
800xAAdmin

Parameter	Value	Type	Unit	Min	Max
Copy unconnected channels	Inputs/Outputs	enum			
Latched hardware state	Enabled	enum			
System alarms on hardware unit	Enabled	enum			
Simple events on hardware units	Enabled	enum			
HwStatus update cycle time	1000	dint	ms	500	30000
Clamp Analog out values	true	bool			

6.5 Test Mode & SoftController

6.5.1 Test Mode

Without downloading to a real controller or a SoftController, you may test your application by using the test mode. In CBM, select “Tools > Test mode”.

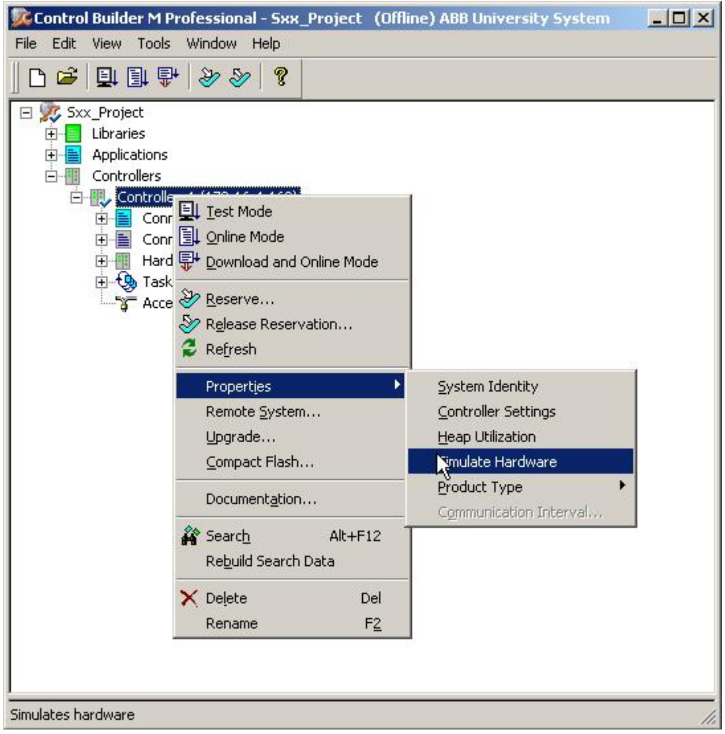
6.5.2 How to use the SoftController

You may also test your application and even your hardware by downloading to a SoftController running on your PC. It is also possible to use the SoftController for communication purposes, for example, COMLI or MODBUS.

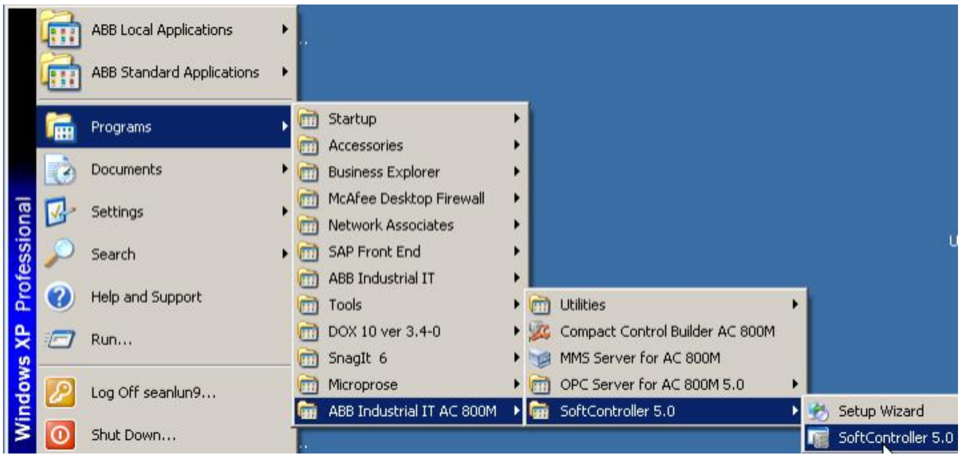
1. Set the controller’s system identity you want to test to your PC’s IP address:2 (such as 172.16.0.1:2).

   NOTE! **It is also possible to use the loop back adapter address as 127.0.0.1:2.**

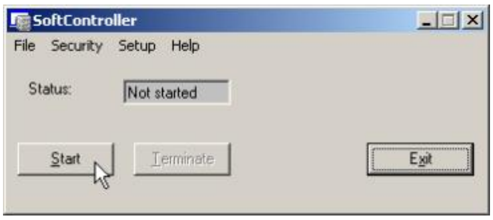
2. Right click the controller, select “Properties > Simulate Hardware”



3. Start the SoftController by double-click the icon on the desktop or choose the path below:

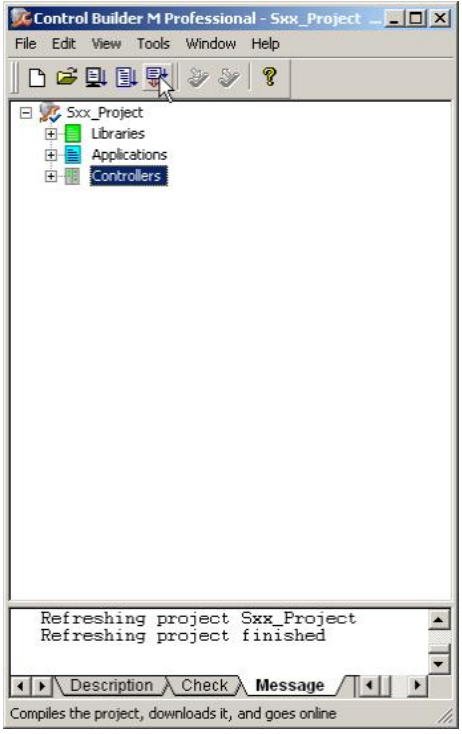


4. Click “Start”.



Status: Not started

5. Choose “Download project and Go online”.



```
Refreshing project Sxx_Project
Refreshing project finished
```

Compiles the project, downloads it, and goes online

Exercise 6.1 Downloading

6.1.1 Description

Perform a download of the project to the controller and set various Control Builder M options.

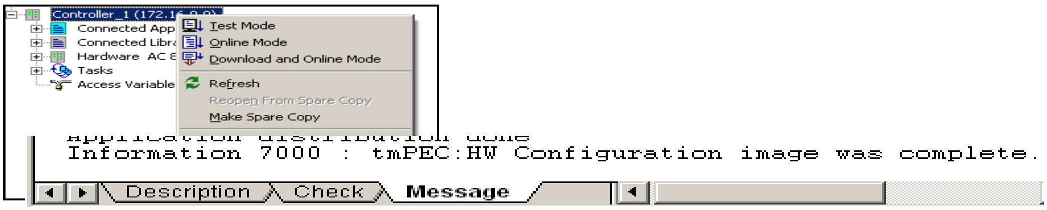
6.1.2 Legend

>	Indicates when you go from one menu to a sub-menu
Italic	Indicates object and file names
“ “	Indicates dialog box buttons, tabs, menus etc.
Bold	Indicates important topics
☞	Indicates start/explanation of student activity

6.1.3 Exercise Steps

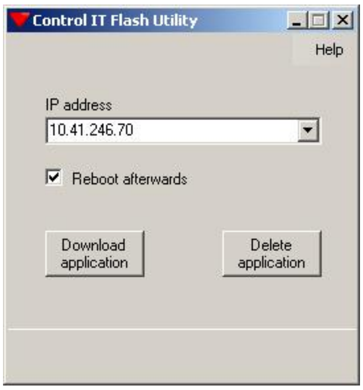
6.1.3.1 Special Case PEC!

For the AC800PEC use the Flash Utility tool to download. The first step is right click on the controller and “make” the data for the compact flash.



```
Application distribution done
Information 7000 : tmPEC:HW Configuration image was complete.
```

The second step is downloading the file to the PEC



IP address: 10.41.246.70

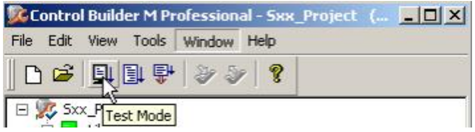
☑ Reboot afterwards

6.1.3.2 Download the Project to the Controller (only RAM)

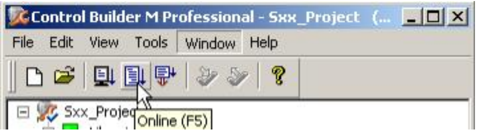
At the moment, our programming consists only of a single application, *Sxx_ReactorApp*, with some default diagnostic program running under the application (*Program3*). This will be sufficient to use for testing our ability to download.

Three icons are located on Control Builder M's toolbar (and also on the menu) which relate to downloading. Place your cursor over each of them momentarily to get a tool-tip text what each of the icons does.

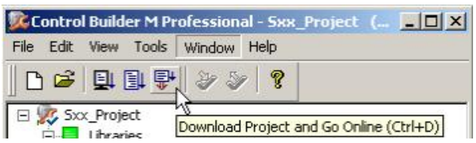
"Test Mode"
Compiles the application and the application executes in the local workstation. The application is NOT downloaded to the physical controller. Test mode is used for checking software before committing it to a controller.



"On-line"
CBM communicates directly with the controller to allow viewing of "live" or "on-line" data. This may only be done if the current application in the controller is the same as that in Control Builder. No download occurs; the Control Builder just connects to the controller and its application.

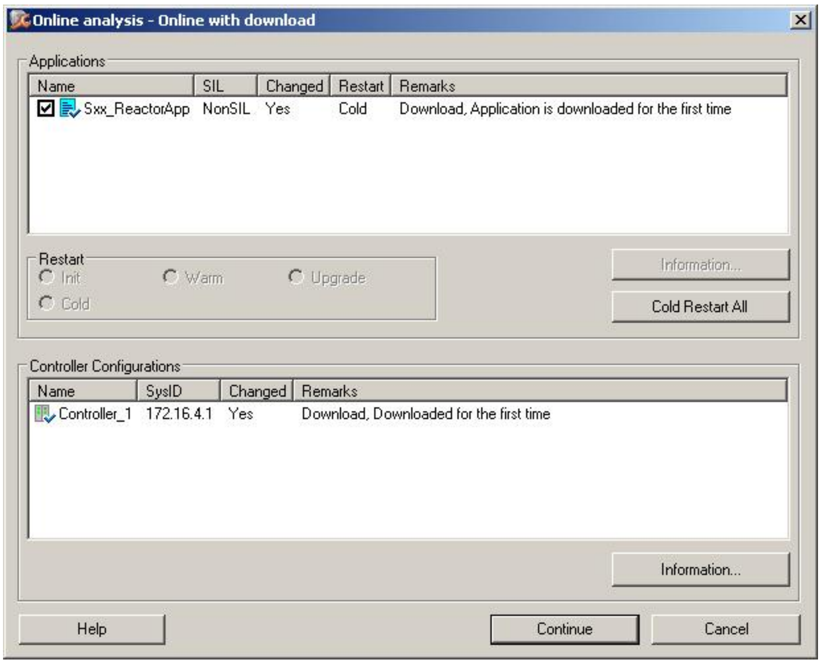


"Download to Ram only and Go Online"
Causes a new compilation of the project files and downloads the compiled code to the controller. After this is done, Control Builder will be placed into the "online" mode as described above. The menu will provide you with the option to perform a coldstart of the new code and possibly a warmstart. For now, just accept the default start type.


Control Builder M Professional - Sxx_Project (...
File
Edit
View
Tools
Window
Help
Sxx_Project
Libraries
Download Project and Go Online (Ctrl+D)

☞ Because this is the first time that the controller has been configured, you need to do a "**Download and Go Online"**.

The Online analysis window will compare the application code and hardware settings in the actual controller and Control Builder.



Online analysis - Online with download

Applications

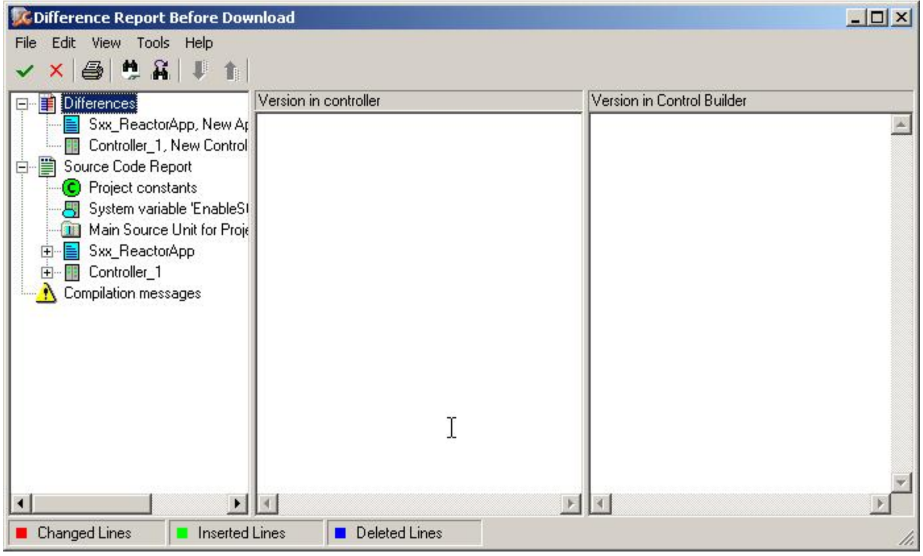
Name	SIL	Changed	Restart	Remarks
☑ Sxx_ReactorApp	NonSIL	Yes	Cold	Download, Application is downloaded for the first time

Restart: ○ Init ○ Warm ○ Upgrade ○ Cold

Controller Configurations

Name	SysID	Changed	Remarks
Controller_1	172.16.4.1	Yes	Download, Downloaded for the first time

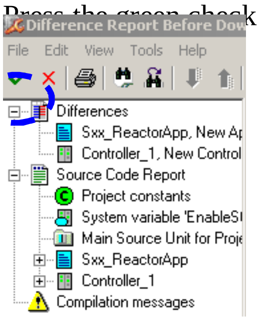
☞ During the download, you will be presented with the “Difference Report.” To view an item of the report, double click on the item in the left-hand panel. The report will be presented in the right hand pane.



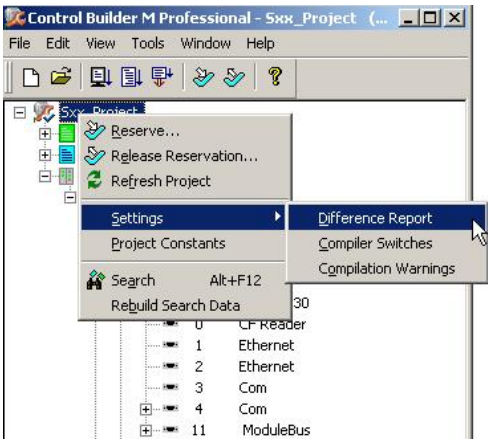
Difference Report Before Download

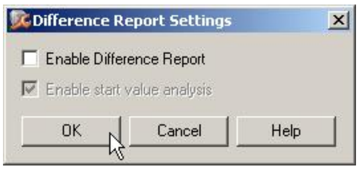
- Differences
  - Sxx_ReactorApp, New Ap
  - Controller_1, New Control
- Source Code Report
  - Project constants
  - System variable 'EnableSt
  - Main Source Unit for Proje
  - Sxx_ReactorApp
  - Controller_1
- Compilation messages

☞ Press the green check-box icon to continue when finished viewing the report.


Difference Report Before Dow
File Edit View Tools Help
Differences
Sxx_ReactorApp, New Ap
Controller_1, New Control
Source Code Report
Project constants
System variable 'EnableSt
Main Source Unit for Proje
Sxx_ReactorApp
Controller_1
Compilation messages

☞ To disable the Difference Report, right click on the project and select **Settings** > **Difference Report**. Remove the checkbox next to Enable Difference Report and click “OK”.

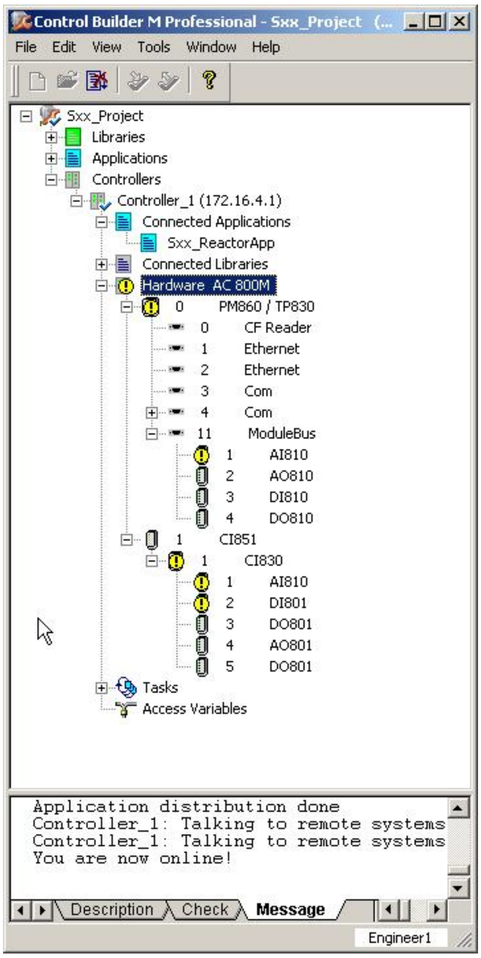



Difference Report Settings

☐ Enable Difference Report

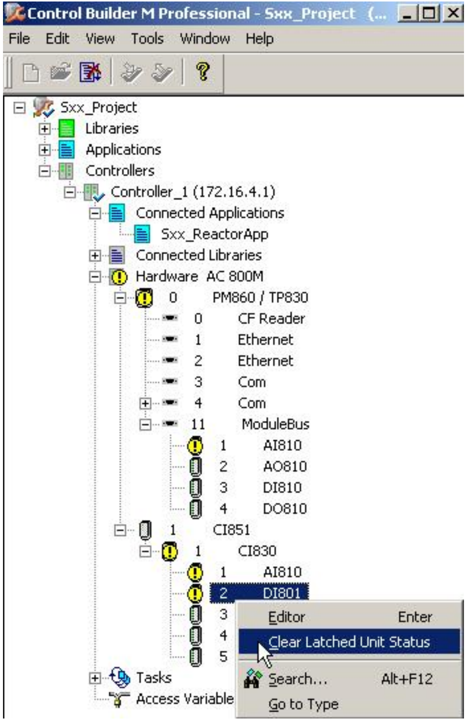
☑ Enable start value analysis

☞ From the “Tools” menu or using the icon on the toolbar, download to the controller and go online.

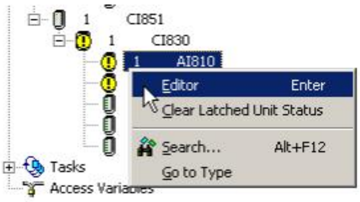


```
Application distribution done
Controller_1: Talking to remote systems
Controller_1: Talking to remote systems
You are now online!
```

☞ Examine any hardware warnings (indicated in yellow on the project tree). First right click on the warning and select “Clear Latched Unit Status”.

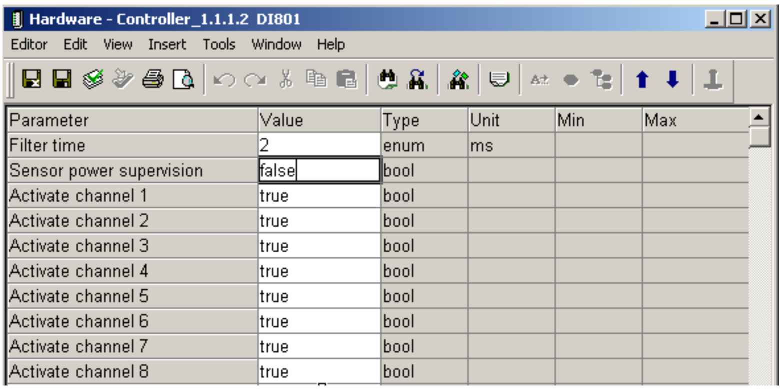


☞ If the warnings do not clear, select the module editor and try to determine the cause of the warning. For Analog Inputs, the warnings may be due to a channel **Underflow** which means the input value is less than the configured minimum.



The **DI801** module can be wired such that channel 16 is used to monitor the power for the digital input signals. Because we are not using that function, it is necessary to turn it off in software. Select the editor for the module and change the value of the parameter **"Sensor Power Supervision"** to **"False"**.

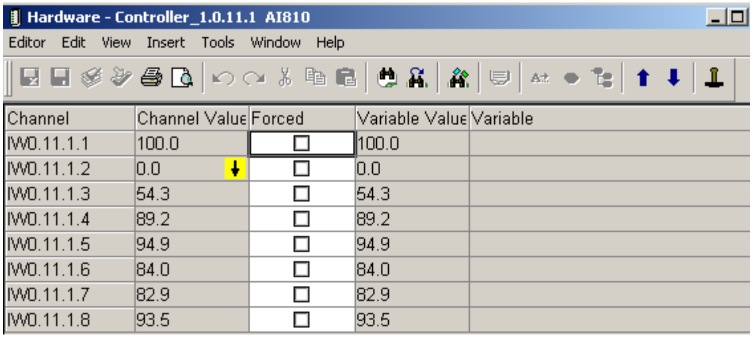
NOTE! Control Builder must be offline to make any changes.


Hardware - Controller_1.1.1.2 DI801
Editor Edit View Insert Tools Window Help

Parameter	Value	Type	Unit	Min	Max
Filter time	2	enum	ms		
Sensor power supervision	false	bool			
Activate channel 1	true	bool			
Activate channel 2	true	bool			
Activate channel 3	true	bool			
Activate channel 4	true	bool			
Activate channel 5	true	bool			
Activate channel 6	true	bool			
Activate channel 7	true	bool			
Activate channel 8	true	bool			

Download again to the controller.

Repeat this for the AI810 module and manipulate the potentiometers on the I/O Panel.


Hardware - Controller_1.0.11.1 AI810
Editor Edit View Insert Tools Window Help

Channel	Channel Value	Forced	Variable Value	Variable
IW0.11.1.1	100.0	□	100.0	
IW0.11.1.2	0.0 ↓	□	0.0	
IW0.11.1.3	54.3	□	54.3	
IW0.11.1.4	89.2	□	89.2	
IW0.11.1.5	94.9	□	94.9	
IW0.11.1.6	84.0	□	84.0	
IW0.11.1.7	82.9	□	82.9	
IW0.11.1.8	93.5	□	93.5	

Exercise 6.2 Writing Simple Code

6.2.1 Description

Write some simple logic code using **Structured Text**. Declare variables and connect them to the input and output channels in the I/O system.

Test the code in online mode and observe the status of simple Boolean variables. Force inputs and outputs using the hardware editor of the relevant I/O module.

6.2.2 Legend

>	Indicates when you go from one menu to a sub-menu
Italic	Indicates object and file names
“ “	Indicates dialog box buttons, tabs, menus etc.
Bold	Indicates important topics
☞	Indicates start/explanation of student activity

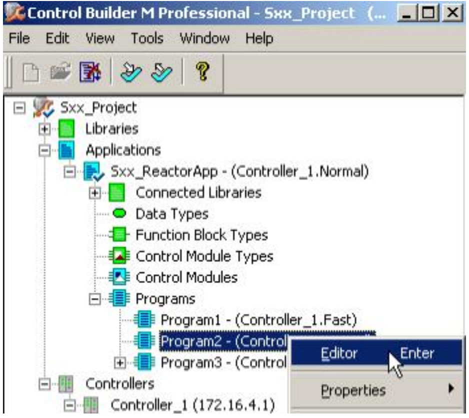
6.2.3 Exercise Steps

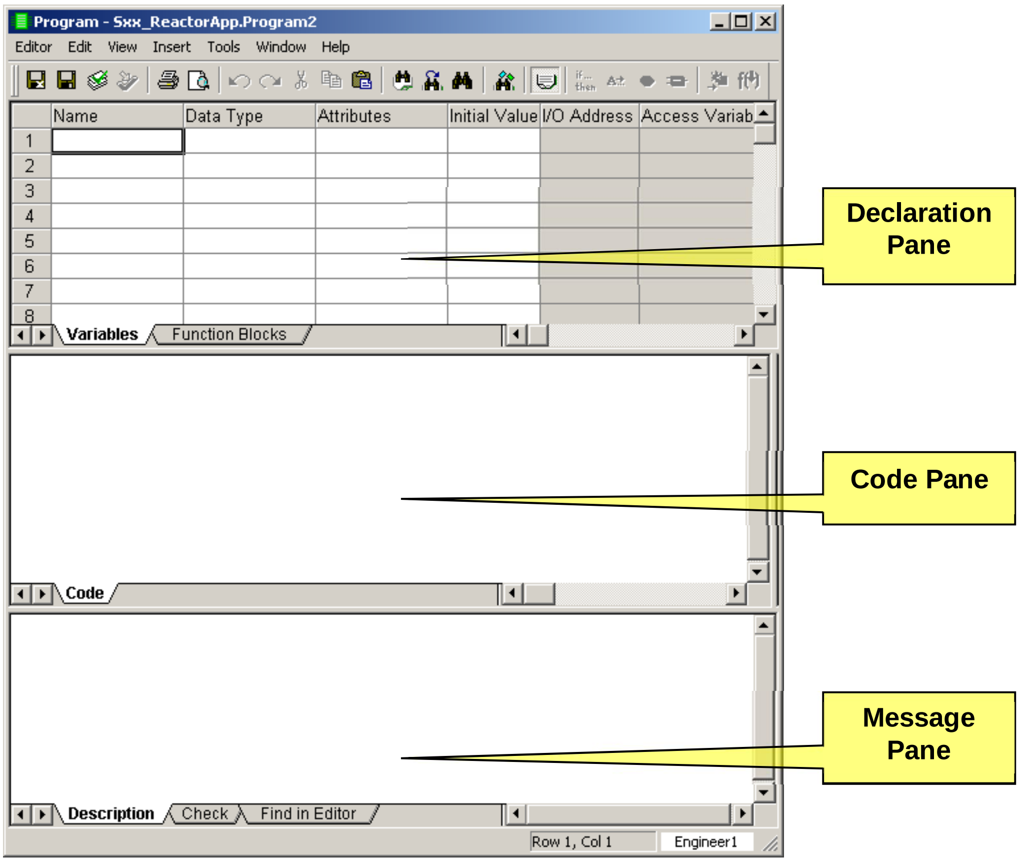
We will be writing some very simple and basic code to run in *Program2* in the *Sxx_ReactorApp* application as an introduction to using the POU editor functions.

NOTE! The editor must be in on off-line mode!

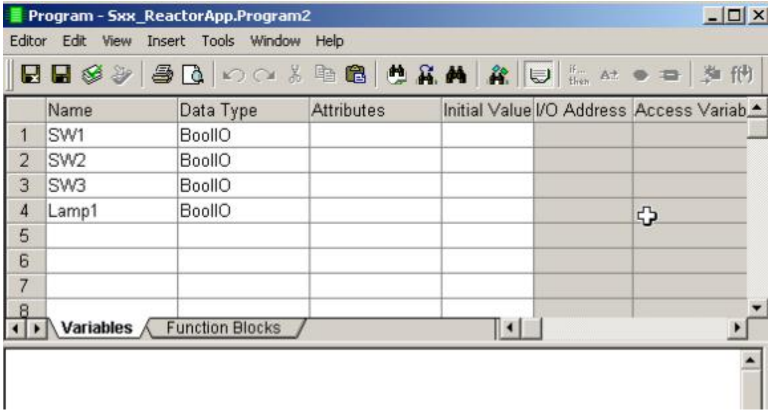
6.2.3.1 Declare Variables

☞ Open ***Program2*** in application *Sxx_ReactorApp*.





☞ We will need to declare variables to hold the values represented by our I/O points. Declare 4 **BoolIO** variables called *SW1*, *SW2*, *SW3*, and *Lamp1* with no attributes.

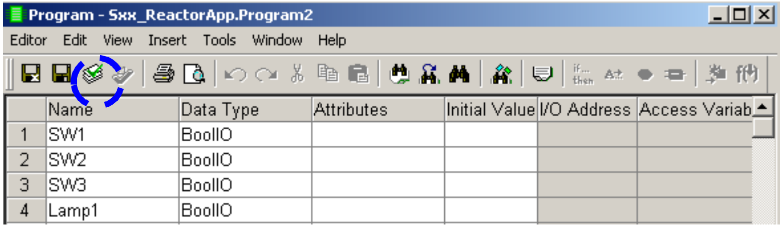

Program - Sxx_ReactorApp.Program2
Editor Edit View Insert Tools Window Help
Variables
Function Blocks

	Name	Data Type	Attributes	Initial Value	I/O Address	Access Variab
1	SW1	BoolIO				
2	SW2	BoolIO				
3	SW3	BoolIO				
4	Lamp1	BoolIO				
5						
6						
7						
8						

NOTE! **Because these values are directly associated with I/O, no retain-type attributes will be set.**

Entries in the “Data Type” and “Attributes” columns may be typed in directly, if known, or a pick-list can be obtained by placing the cursor in the desired cell in the editor and pressing **<CTRL-J>**.

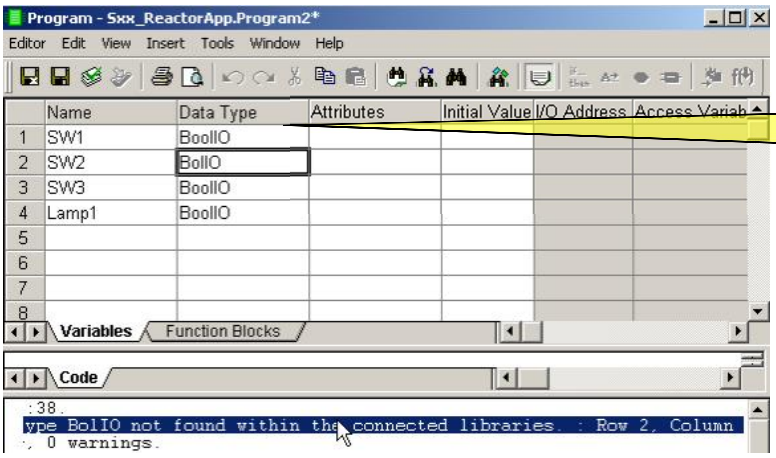
☞ After making changes check the Program for any errors.


Program - Sxx_ReactorApp.Program2
Editor Edit View Insert Tools Window Help

	Name	Data Type	Attributes	Initial Value	I/O Address	Access Variab
1	SW1	BoolIO				
2	SW2	BoolIO				
3	SW3	BoolIO				
4	Lamp1	BoolIO				

If Control Builder detects errors, they will be shown in the message (lower) pane in the POU editor. This pane may have to be enabled by selecting “View > Messages Pane” from the menu.

Double-click on an error in the message pane to locate the cursor where the error was detected.


Program - Sxx_ReactorApp.Program2*
Editor Edit View Insert Tools Window Help
Variables
Function Blocks
Code

	Name	Data Type	Attributes	Initial Value	I/O Address	Access Variab
1	SW1	BoolIO				
2	SW2	BolIO				
3	SW3	BoolIO				
4	Lamp1	BoolIO				
5						
6						
7						
8						

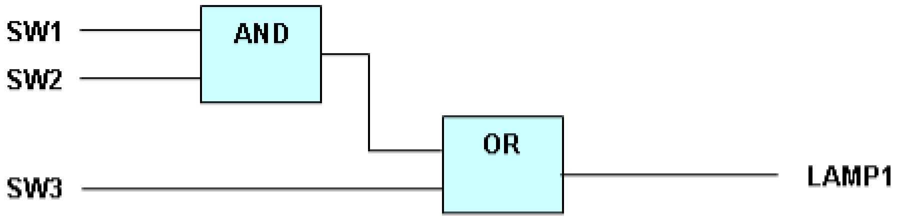
```
 :38.
ype BolIO not found within the connected libraries. : Row 2, Column
·, 0 warnings.
```

Cursor Shown on Error

6.2.3.2 Write Logic

We will need to write some logic in the Code Pane in the editor. Programming will actually be covered in several lessons later in the course. We will be inserting some programming code in the **Structured Text** (ST) language – the default language the editor opened up with.

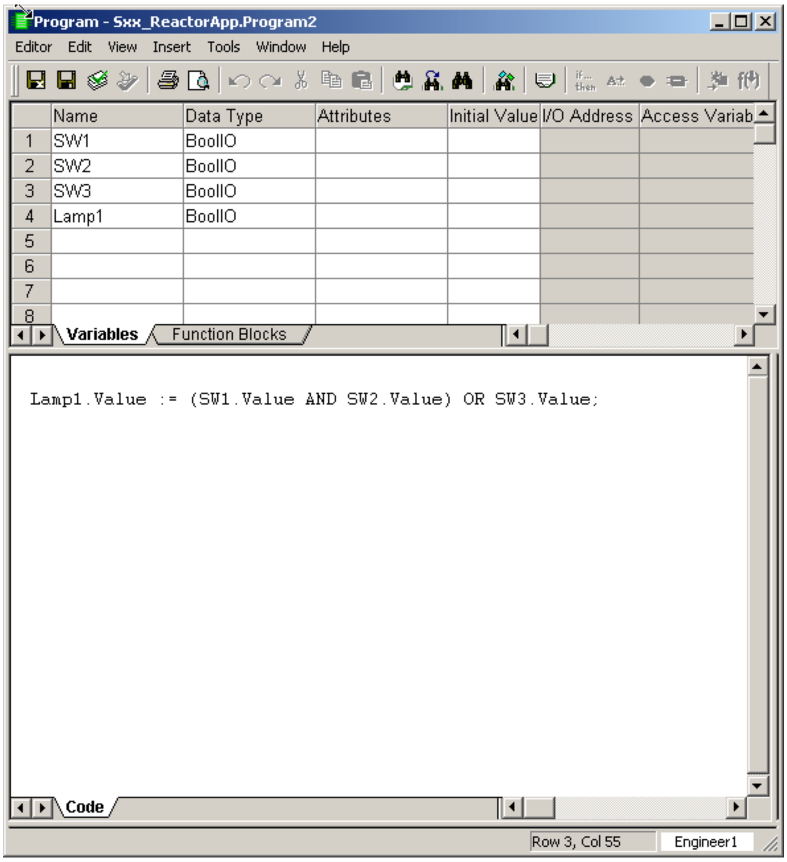
The logic behind our code is very simple. It will evaluate three Digital Inputs and set a Digital Output according to the logic diagram below:



The SW1, SW2, and SW3 variables will be connected to three channels of a digital input module (which are wired to switches) and the Lamp1 variable will be connected to a channel on a digital output module (which is wired to an indicating light).

☞ Type the following line directly in the Code Pane area in the editor. Note that the = sign is actually preceded by a colon and the line must end with a semicolon.

Lamp1.Value := (SW1.Value AND SW2.Value) OR SW3.Value;

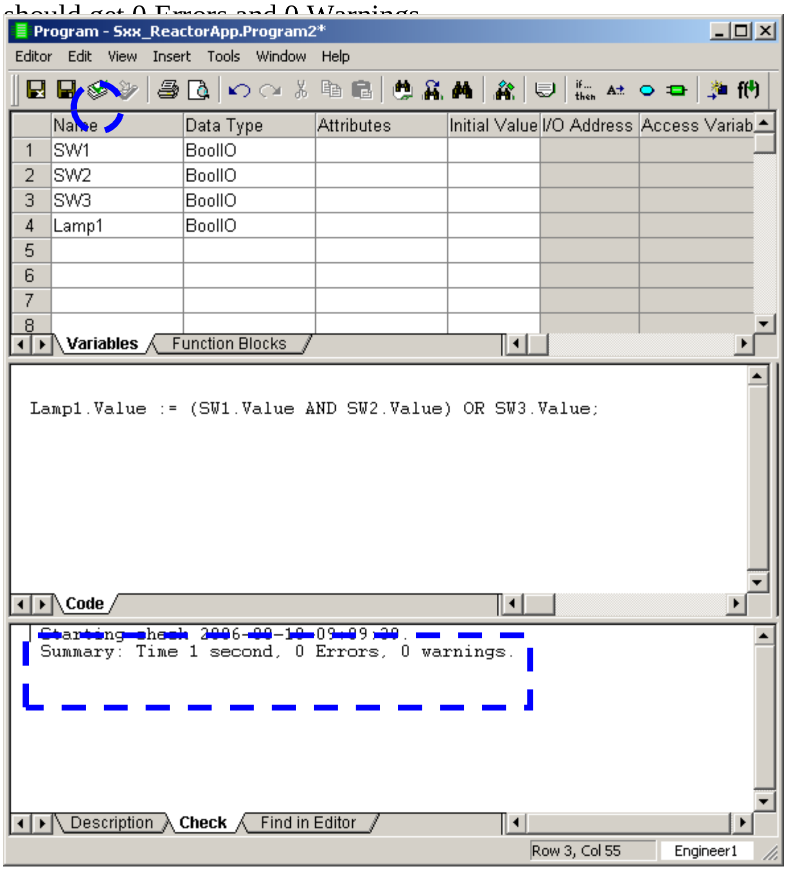


	Name	Data Type	Attributes	Initial Value	I/O Address	Access Variab
1	SW1	BoolIO				
2	SW2	BoolIO				
3	SW3	BoolIO				
4	Lamp1	BoolIO				
5						
6						
7						
8						

```
Lamp1.Value := (SW1.Value AND SW2.Value) OR SW3.Value;
```

Row 3, Col 55   Engineer1

☞ Perform a check of your syntax by clicking on the “Check” icon in the tool bar: You should get 0 Errors and 0 Warnings.



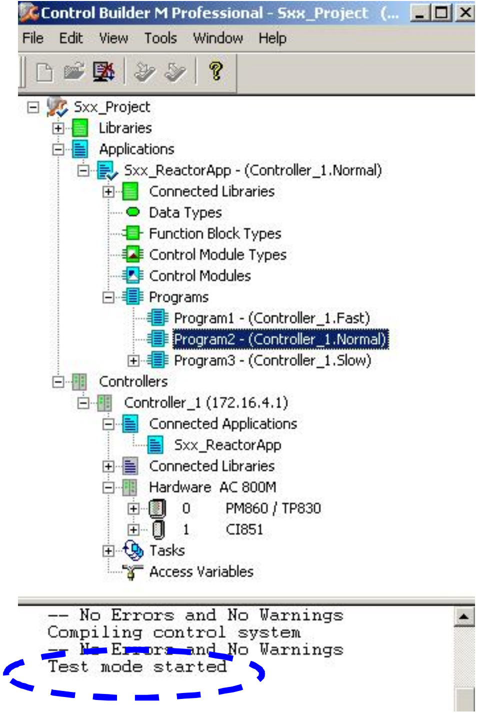
	Name	Data Type	Attributes	Initial Value	I/O Address	Access Variab
1	SW1	BoolIO				
2	SW2	BoolIO				
3	SW3	BoolIO				
4	Lamp1	BoolIO				
5						
6						
7						
8						

```
Lamp1.Value := (SW1.Value AND SW2.Value) OR SW3.Value;
```

```
Starting check 2006-09-19 09:09:39.
Summary: Time 1 second, 0 Errors, 0 warnings.
```

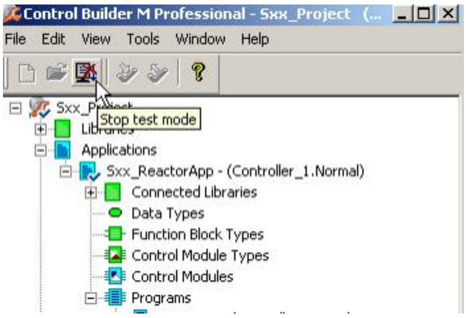
☞ Save and close the POU editor.

☞ Do an off-line test of the logic by taking Control Builder to Test Mode. Click the "Continue" button for the On Line Change analysis dialog and observe that Control Builder enters the "Test" mode by checking the Message Pane in Control Builder.



```
-- No Errors and No Warnings
Compiling control system
-- No Errors and No Warnings
Test mode started
```

☞ Take Control Builder to the Off-line mode by pressing the "Stop Test Mode" icon on the toolbar.

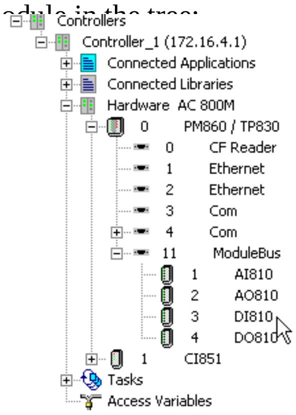


6.2.3.3 Connect Variables to I/O Channels

Once variables have been defined (and saved) in a POU editor, they can then become available to connect to the actual hardware channel itself. This is called "I/O Allocation". I/O allocation is configured in the editor window for each specific I/O module.

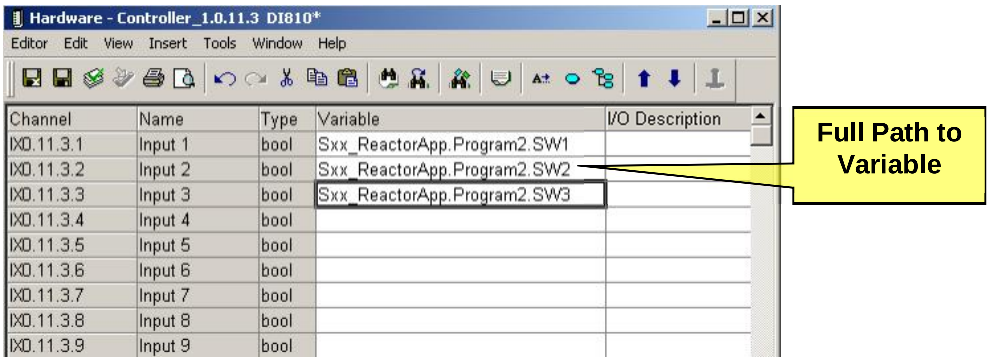
Right click on the I/O module to open the Hardware editor and selecting "Editor" or double-click the module object itself. The settings available on each editor will be different depending upon the hardware module type.

☞ Open up the hardware structure in the Control Builder and double click on the DI810 module in the tree:



```
Controllers
  Controller_1 (172.16.4.1)
    Connected Applications
    Connected Libraries
    Hardware AC 800M
      0   PM860 / TP830
        0   CF Reader
        1   Ethernet
        2   Ethernet
        3   Com
        4   Com
        11  ModuleBus
          1   AI810
          2   AO810
          3   DI810
          4   DO810
      1   CI851
    Tasks
    Access Variables
```

☞ Connect the three switches on the DI810 module to the *SW1*, *SW2*, and *SW3* variables you created in "Program2". Use the icons in the upper right corner of the editor to choose variables. Save and close the editor when finished.

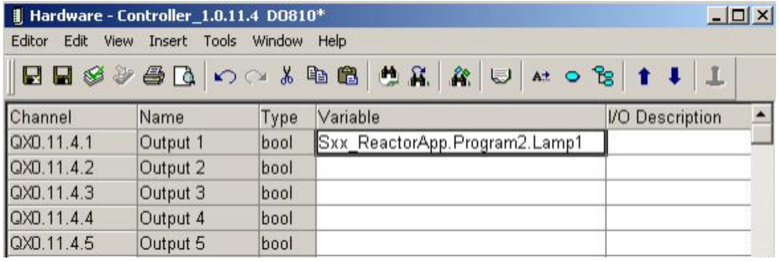


Hardware - Controller_1.0.11.3 DI810*
Editor Edit View Insert Tools Window Help

Channel	Name	Type	Variable	I/O Description
IX0.11.3.1	Input 1	bool	Sxx_ReactorApp.Program2.SW1	
IX0.11.3.2	Input 2	bool	Sxx_ReactorApp.Program2.SW2	
IX0.11.3.3	Input 3	bool	Sxx_ReactorApp.Program2.SW3	
IX0.11.3.4	Input 4	bool		
IX0.11.3.5	Input 5	bool		
IX0.11.3.6	Input 6	bool		
IX0.11.3.7	Input 7	bool		
IX0.11.3.8	Input 8	bool		
IX0.11.3.9	Input 9	bool		

Full Path to Variable

☞ Connect the first channel on the DO810 module to the *Lamp1* variable you created in *Program2*. Check for errors and then save and close the editor when finished.



Hardware - Controller_1.0.11.4 DO810*
Editor Edit View Insert Tools Window Help

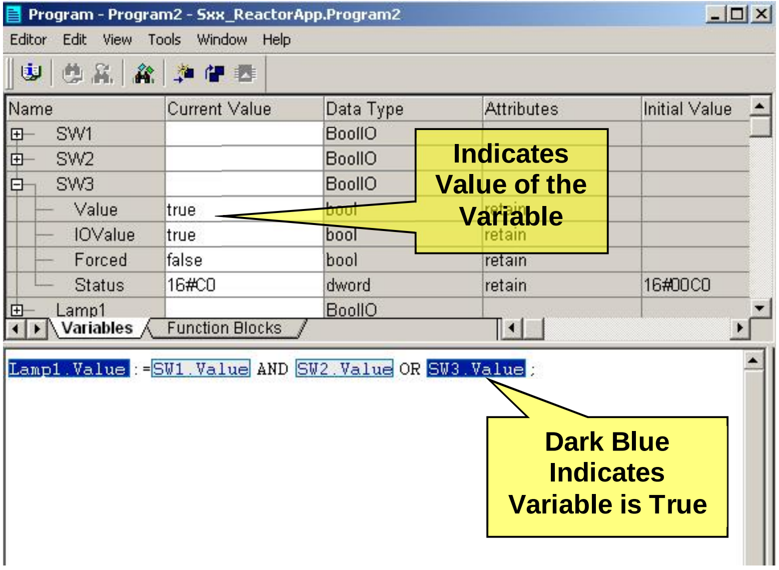
Channel	Name	Type	Variable	I/O Description
QX0.11.4.1	Output 1	bool	Sxx_ReactorApp.Program2.Lamp1	
QX0.11.4.2	Output 2	bool		
QX0.11.4.3	Output 3	bool		
QX0.11.4.4	Output 4	bool		
QX0.11.4.5	Output 5	bool		

6.2.3.4 Download and Go Online

☞ Download your project to the controller.

6.2.3.5 Viewing Code in Online Mode

☞ Ensure Control Builder is in the Online mode and open the POU editor for *Program 2* and observer your logic in on-line mode. Manipulate the three switches and observe the light on the I/O panel while looking at your logic.



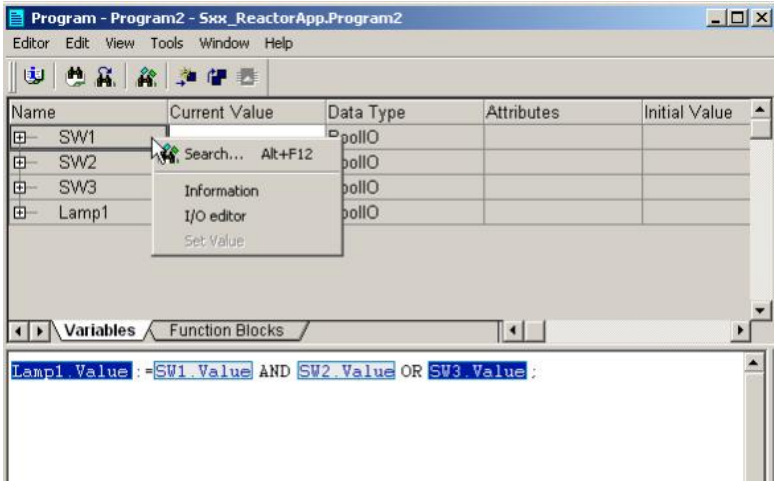
Name	Current Value	Data Type	Attributes	Initial Value
SW1		BoolIO		
SW2		BoolIO		
SW3		BoolIO		
Value	true	bool	retain	
IOValue	true	bool	retain	
Forced	false	bool	retain	
Status	16#C0	dword	retain	16#00C0
Lamp1		BoolIO		

```
Lamp1.Value:=SW1.Value AND SW2.Value OR SW3.Value;
```

Indicates Value of the Variable

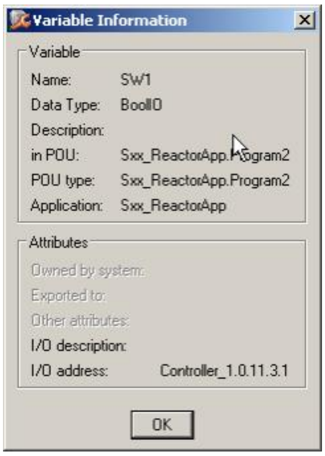
Dark Blue Indicates Variable is True

☞ Right click on the variable in the Declaration Pane to get information about the variable and also to open the I/O editor.



Name	Current Value	Data Type	Attributes	Initial Value
SW1		BoolIO		
SW2		BoolIO		
SW3		BoolIO		
Lamp1		BoolIO		

```
Lamp1.Value:=SW1.Value AND SW2.Value OR SW3.Value;
```



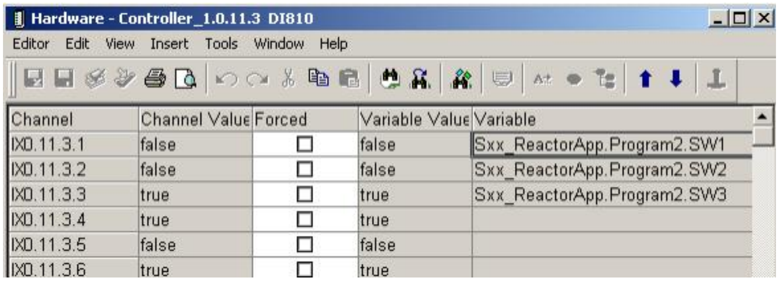
Variable Information

Variable

Name: SW1
Data Type: BoolIO
Description:
in POU: Sxx_ReactorApp.Program2
POU type: Sxx_ReactorApp.Program2
Application: Sxx_ReactorApp

Attributes

Owned by system:
Exported to:
Other attributes:
I/O description:
I/O address: Controller_1.0.11.3.1

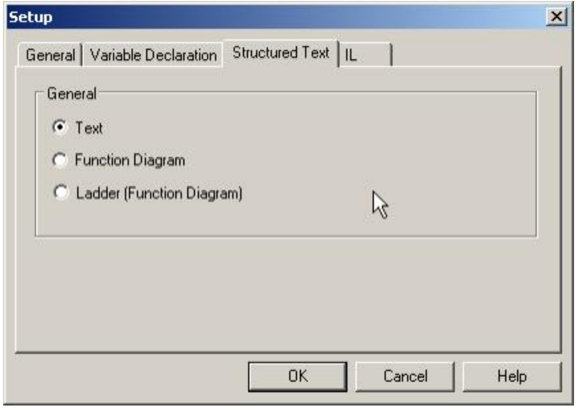


Hardware - Controller_1.0.11.3 DI810

Channel	Channel Value	Forced	Variable Value	Variable
IX0.11.3.1	false	☐	false	Sxx_ReactorApp.Program2.SW1
IX0.11.3.2	false	☐	false	Sxx_ReactorApp.Program2.SW2
IX0.11.3.3	true	☐	true	Sxx_ReactorApp.Program2.SW3
IX0.11.3.4	true	☐	true	
IX0.11.3.5	false	☐	false	
IX0.11.3.6	true	☐	true	

6.2.3.6 Change the Online View of Structured Text (ST)

If your code is written in ST, you can alter the views of your logic to either Function Diagram or Ladder (other available programming languages).
Select **Tools** > **Setup** from the program editor menu bar.

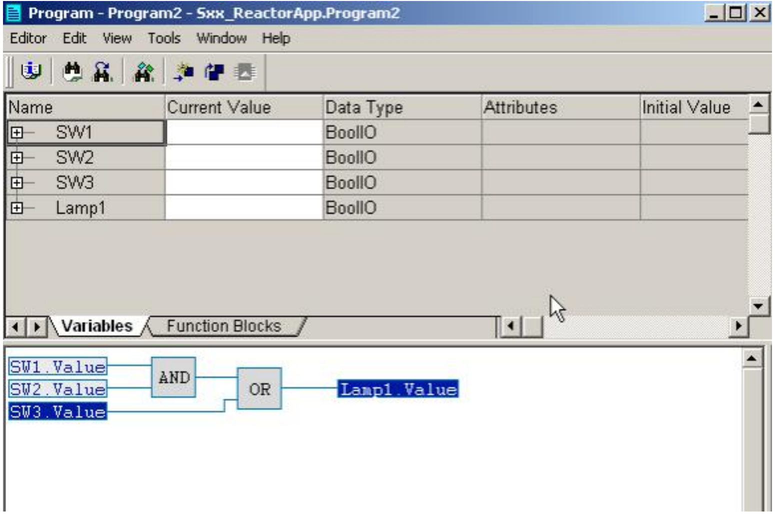


Setup

General

● Text
○ Function Diagram
○ Ladder (Function Diagram)

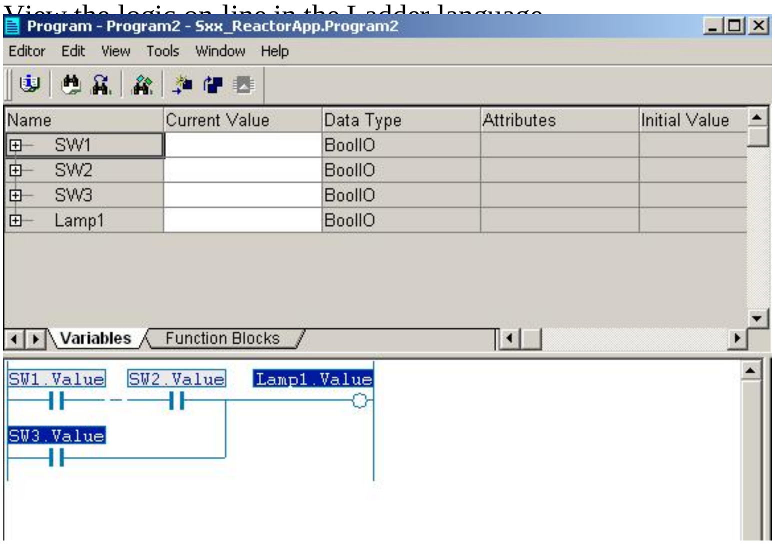
☞ View the logic on-line in the Function Diagram language.



Program - Program2 - Sxx_ReactorApp.Program2

Name	Current Value	Data Type	Attributes	Initial Value
⊞ SW1		BoolIO		
⊞ SW2		BoolIO		
⊞ SW3		BoolIO		
⊞ Lamp1		BoolIO		

☞ View the logic on line in the Ladder language.



Program - Program2 - Sxx_ReactorApp.Program2
Editor Edit View Tools Window Help
Variables
Function Blocks
SW1.Value
SW2.Value
Lamp1.Value
SW3.Value

Name	Current Value	Data Type	Attributes	Initial Value
⊞ SW1		BoolIO		
⊞ SW2		BoolIO		
⊞ SW3		BoolIO		
⊞ Lamp1		BoolIO		

☞ Change the view back to “Text”.

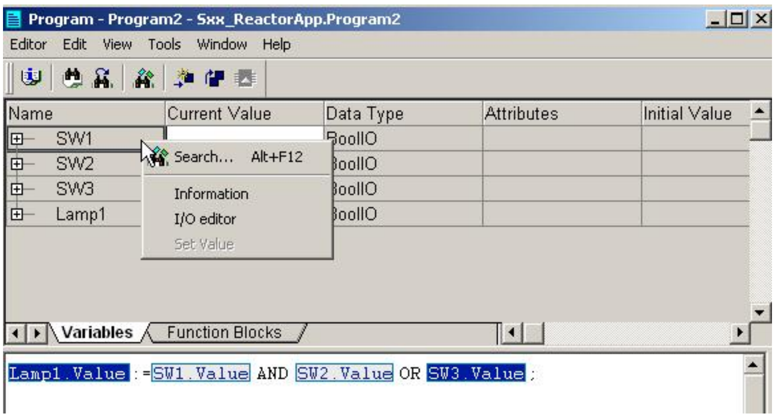
6.2.3.7 Force I/O Signals

I/O may be forced from the “Status” tab in the Hardware editor for the particular I/O module. The editor will have two columns of interest.

- “Channel Value” indicates the actual measured value at the I/O module
- “Variable Value” indicates the value of the variable in the program logic that the I/O channel is connected to

These two values will be equal if the channel is not forced. For input modules, you can change the value of the variable being written to by the I/O module. For output modules, you can force the channel’s output value.

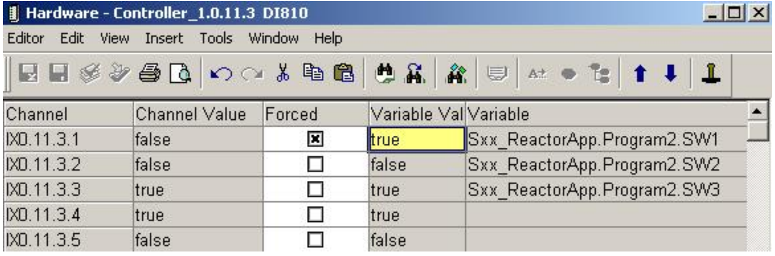
☞ Right-click on the *SW1* variable and select “I/O Editor” to open the hardware editor for the DI810 module.


Program - Program2 - Sxx_ReactorApp.Program2
Editor Edit View Tools Window Help
Search... Alt+F12
Information
I/O editor
Set Value
Variables
Function Blocks

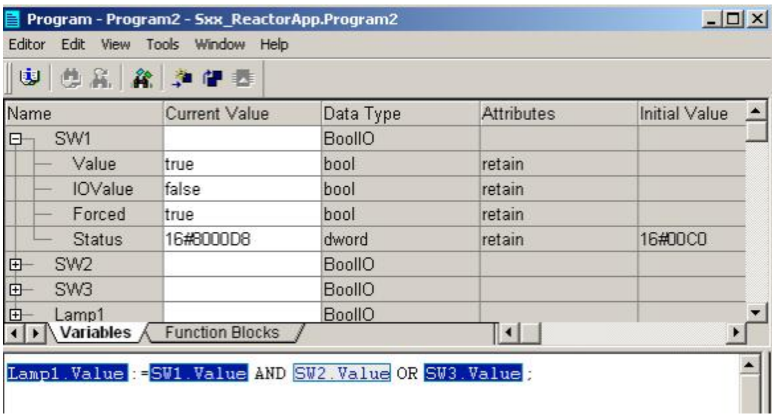
Name	Current Value	Data Type	Attributes	Initial Value
⊞ SW1		BoolIO		
⊞ SW2		BoolIO		
⊞ SW3		BoolIO		
⊞ Lamp1		BoolIO		

```
Lamp1.Value := SW1.Value AND SW2.Value OR SW3.Value ;
```

☞ Ensure the “Status” tab is selected at the bottom of the hardware editor and force one or more variables. Observe the resulting changes in your Program2 editor’s code pane as you force and release forces on variables.


Hardware - Controller_1.0.11.3 DI810
Editor Edit View Insert Tools Window Help

Channel	Channel Value	Forced	Variable Val	Variable
IX0.11.3.1	false	☒	true	Sxx_ReactorApp.Program2.SW1
IX0.11.3.2	false	☐	false	Sxx_ReactorApp.Program2.SW2
IX0.11.3.3	true	☐	true	Sxx_ReactorApp.Program2.SW3
IX0.11.3.4	true	☐	true	
IX0.11.3.5	false	☐	false	


Program - Program2 - Sxx_ReactorApp.Program2
Editor Edit View Tools Window Help
Variables
Function Blocks

Name	Current Value	Data Type	Attributes	Initial Value
⊟ SW1		BoolIO		
Value	true	bool	retain	
IOValue	false	bool	retain	
Forced	true	bool	retain	
Status	16#8000D8	dword	retain	16#00C0
⊞ SW2		BoolIO		
⊞ SW3		BoolIO		
⊞ Lamp1		BoolIO		

```
Lamp1.Value := SW1.Value AND SW2.Value OR SW3.Value ;
```

Solution 6.2 Writing Simple Code

6.2.1 Description

Write some simple logic code using **Structured Text**. Declare variables and connect them to the input and output channels in the IO system.

Test the code in online mode and observe the status of simple Boolean variables. Force inputs and outputs using the hardware editor of the relevant IO module.

6.2.2 Legend

>	Indicates when you go from one menu to a sub-menu
Italic	Indicates object and file names
“ “	Indicates dialog box buttons, tabs, menus etc.
Bold	Indicates important topics
☞	Indicates start/explanation of student activity

6.2.3 Solution Steps

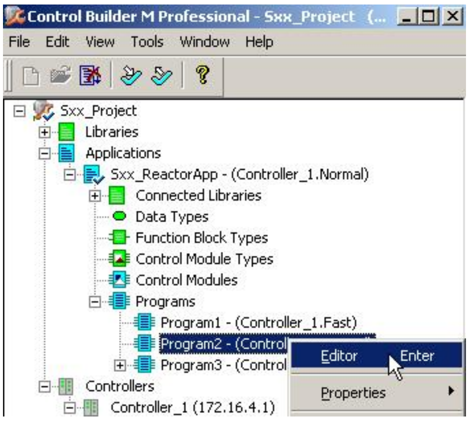
We will be writing some very simple and basic code to run in *Program2* in the *Sxx_ReactorApp* application as an introduction to using the POU editor functions.

NOTE! The editor must be in on off-line mode!

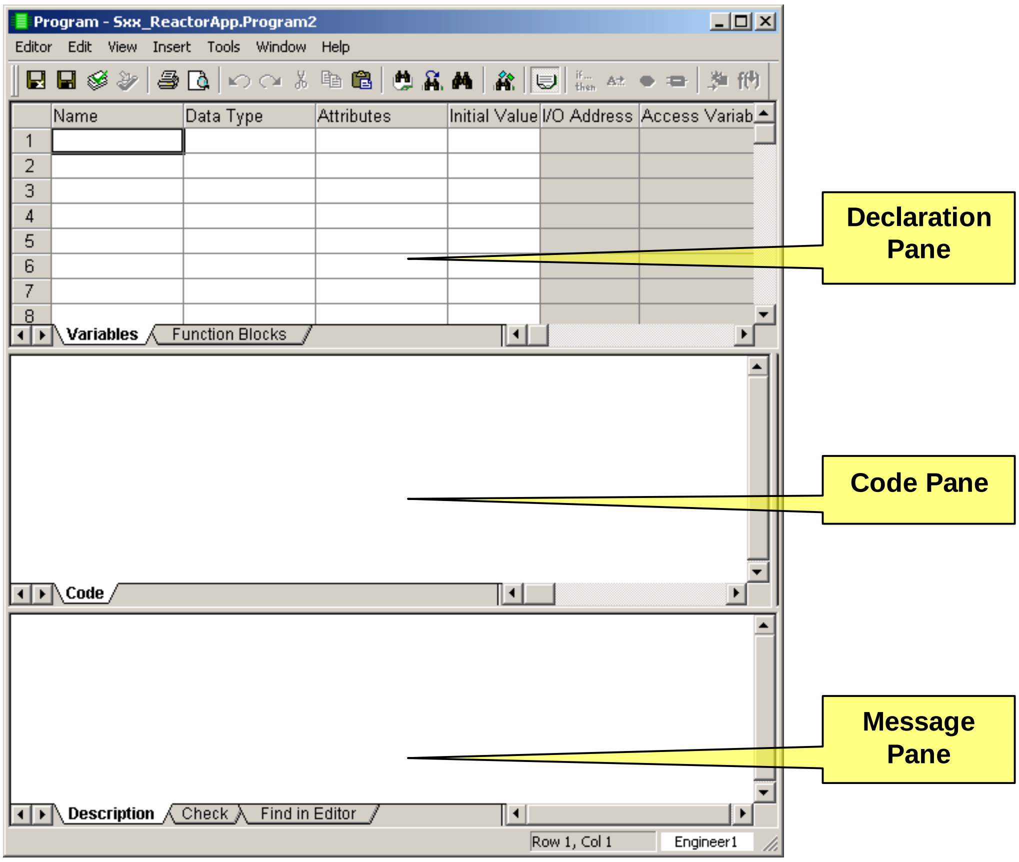
6.2.3.1 Declare Variables

☞ Open ***Program2*** in application *Sxx_ReactorApp*.

1. Make sure that Control Builder M is in offline mode
2. Expand the *Applications* object and select *Program2* located under your *Sxx_ReactorApp* object. You can open the POU for this program either by double-clicking *Program2* or right-clicking and selecting “Editor” from the context menu.



3. Locate the available panes, tabs, and menu items to allow you to navigate in the editor window.



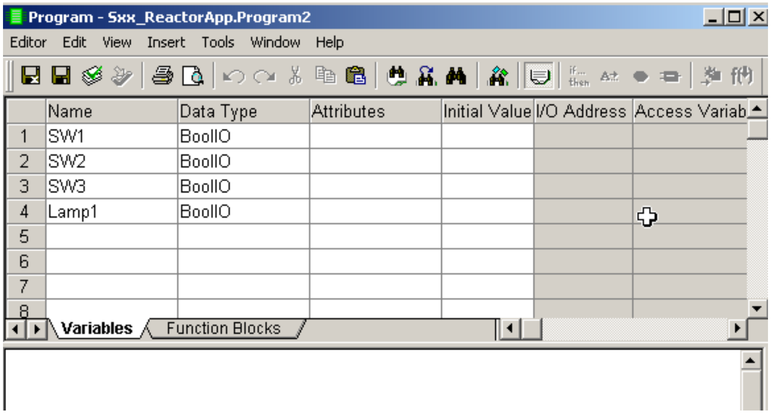
Declaration Pane

Code Pane

Message Pane

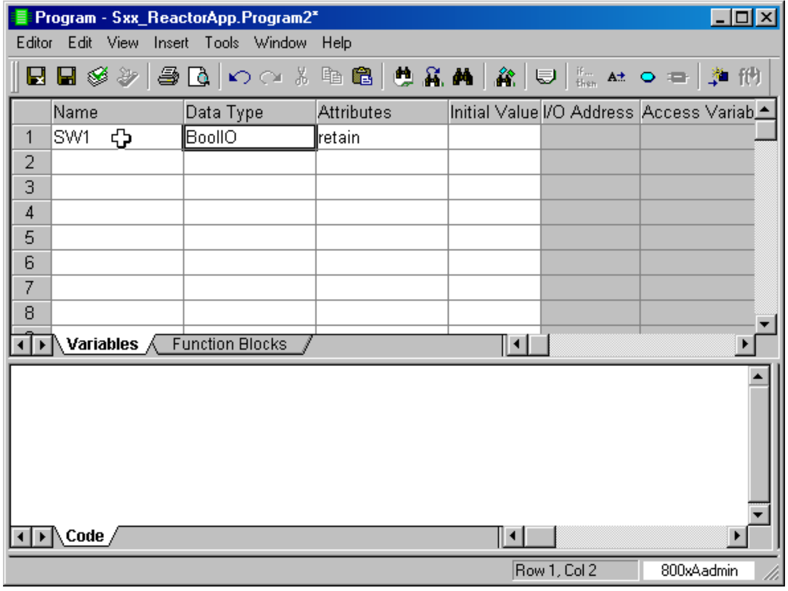
☞ Declare 4 **BoolIO** variables called *SW1*, *SW2*, *SW3*, and *LAMP1* with no attributes.

1. Make sure you are selected to the “Variables” tab in the declaration (upper) pane in the POU editor and enter the names of the variable in the “Name” column.


Program - Sxx_ReactorApp.Program2
Editor Edit View Insert Tools Window Help
Variables
Function Blocks

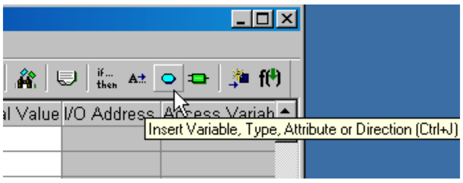
	Name	Data Type	Attributes	Initial Value	I/O Address	Access Variab
1	SW1	BoolIO				
2	SW2	BoolIO				
3	SW3	BoolIO				
4	Lamp1	BoolIO				
5						
6						
7						
8						

2. Place your cursor in the DataType column and press **<CTRL-J>** and select the data type “BoolIO”.

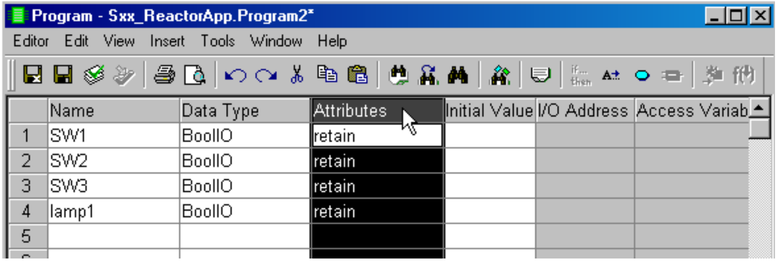

Program - Sxx_ReactorApp.Program2*
Editor Edit View Insert Tools Window Help
Variables
Function Blocks
Code
Row 1, Col 2
800xAadmin

	Name	Data Type	Attributes	Initial Value	I/O Address	Access Variab
1	SW1	BoolIO	retain			
2						
3						
4						
5						
6						
7						
8						

3. Also try selecting the “Insert…” Icon on the upper right corner of the POU editor window.

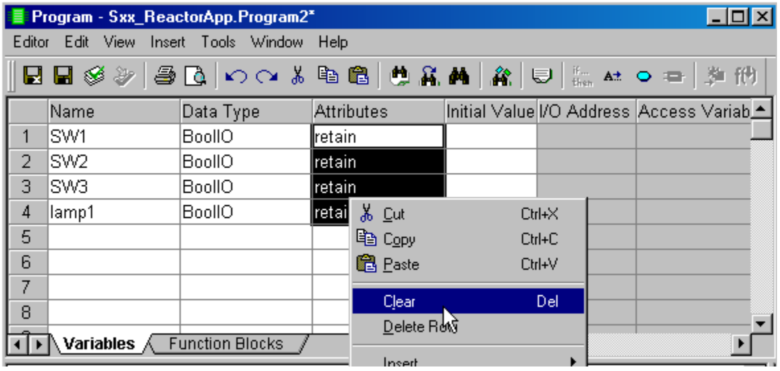

l Value I/O Address Access Variab
Insert Variable, Type, Attribute or Direction (Ctrl+J)

4. Delete the retain attributes which by default will be added when creating variables. Select the entire “Attributes” column by clicking on the column header and delete all entries by pressing the “delete” key on the keyboard.


Program - Sxx_ReactorApp.Program2*
Editor Edit View Insert Tools Window Help

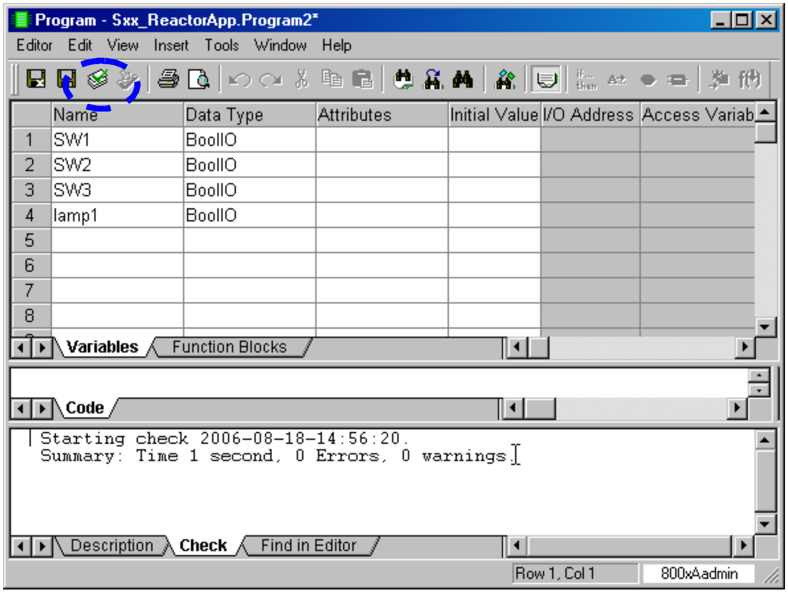
	Name	Data Type	Attributes	Initial Value	I/O Address	Access Variab
1	SW1	BoolIO	retain			
2	SW2	BoolIO	retain			
3	SW3	BoolIO	retain			
4	lamp1	BoolIO	retain			
5						

5. Or mark with the mouse and right click, select “Clear”



	Name	Data Type	Attributes	Initial Value	I/O Address	Access Variab
1	SW1	BoolIO	retain			
2	SW2	BoolIO	retain			
3	SW3	BoolIO	retain			
4	lamp1	BoolIO	retai			
5						
6						
7						
8						

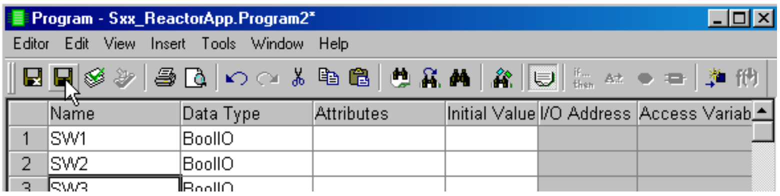
6. Check the POU for errors. In the tab “check” the 0 Errors, 0 warnings



	Name	Data Type	Attributes	Initial Value	I/O Address	Access Variab
1	SW1	BoolIO				
2	SW2	BoolIO				
3	SW3	BoolIO				
4	lamp1	BoolIO				
5						
6						
7						
8						

```
Starting check 2006-08-18-14:56:20.
Summary: Time 1 second, 0 Errors, 0 warnings
```

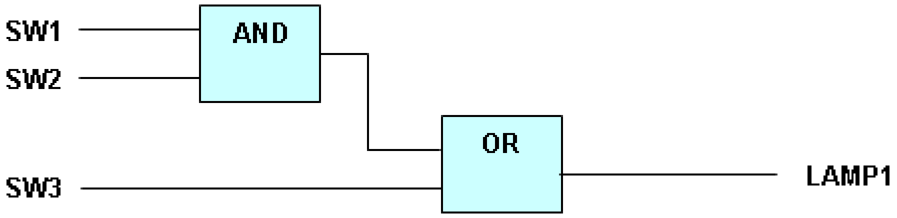
7. Save


Program - Sxx_ReactorApp.Program2*
Editor Edit View Insert Tools Window Help

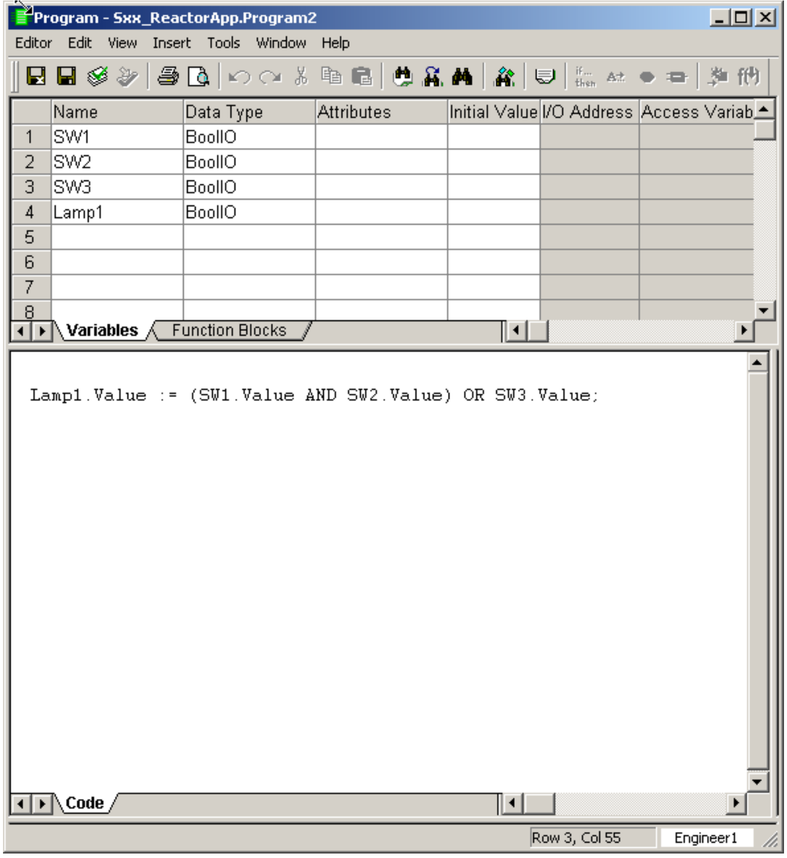
	Name	Data Type	Attributes	Initial Value	I/O Address	Access Variab
1	SW1	BoolIO				
2	SW2	BoolIO				
3	SW3	BoolIO				

6.2.3.2 Write Logic

☞ The logic must valuate three Digital Inputs and set a Digital Output according to the logic diagram
below:



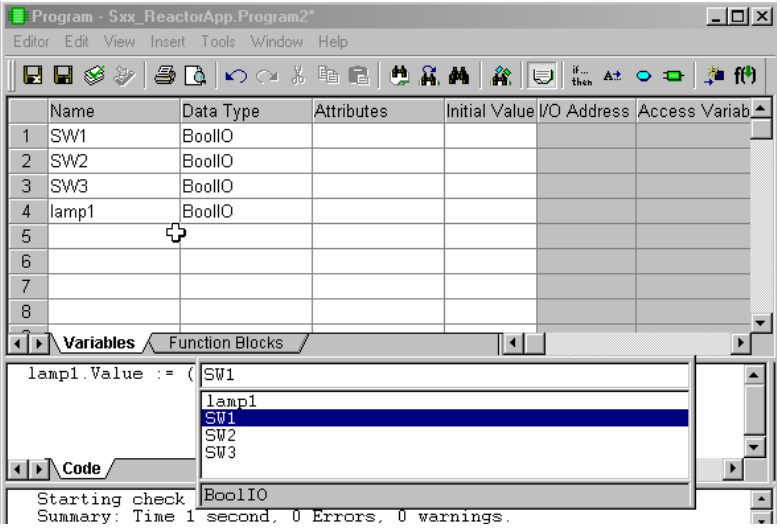
1. Type the following line directly in the Code Pane area in the editor.
Lamp1.Value := (SW1.Value AND SW2.Value) OR SW3.Value;


Program - Sxx_ReactorApp.Program2
Editor Edit View Insert Tools Window Help
Variables
Function Blocks
Code
Row 3, Col 55
Engineer1

	Name	Data Type	Attributes	Initial Value	I/O Address	Access Variab
1	SW1	BoolIO				
2	SW2	BoolIO				
3	SW3	BoolIO				
4	Lamp1	BoolIO				
5						
6						
7						
8						

```
Lamp1.Value := (SW1.Value AND SW2.Value) OR SW3.Value;
```

2. The already defined variables can be marked from a list using keyboard Ctrl J
   The following keyboard code can be used:
   Ctrl j, **arrow down**, select **SW1**, **Enter**,
   .(dot), **arrow down**, select **Value**, **Enter**.
   Arrow down can be replaced by typing the first letter in the item of the list


Program - Sxx_ReactorApp.Program2*
Editor Edit View Insert Tools Window Help
Variables
Function Blocks
Code

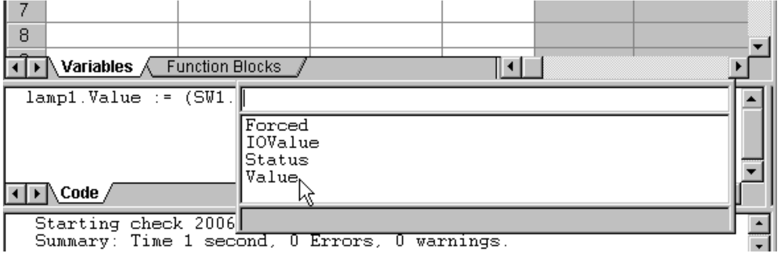
	Name	Data Type	Attributes	Initial Value	I/O Address	Access Variab
1	SW1	BoolIO				
2	SW2	BoolIO				
3	SW3	BoolIO				
4	lamp1	BoolIO				
5						
6						
7						
8						

```
lamp1.Value := (
```

```
SW1
lamp1
SW1
SW2
SW3
BoolIO
```

```
Starting check
Summary: Time 1 second, 0 Errors, 0 warnings.
```

3. Select the keyboard . (dot) behind the SW1 and the Value will be browsed

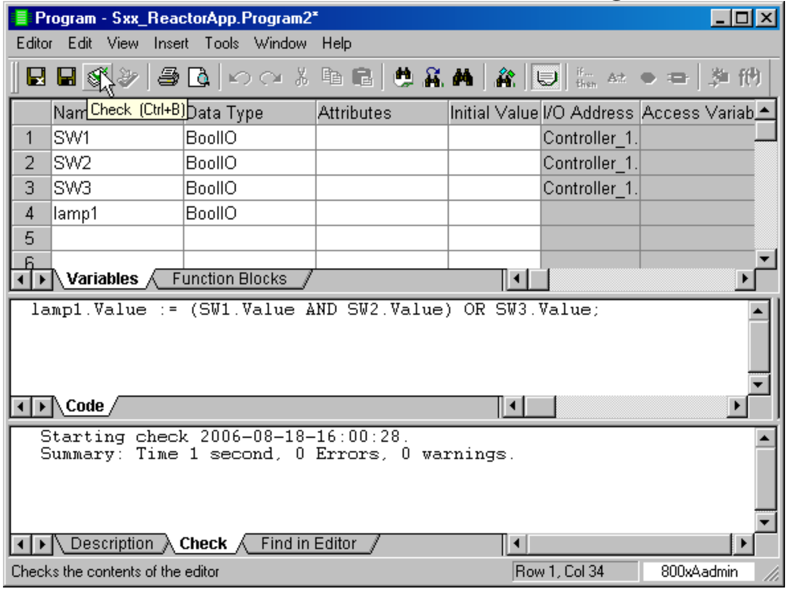


```
lamp1.Value := (SW1.
```

```
Forced
IOValue
Status
Value
```

```
Starting check 2006
Summary: Time 1 second, 0 Errors, 0 warnings.
```

4. Select the ‘Check’ Icon in the tool bar: You should get 0 Errors and 0 Warnings.


Program - Sxx_ReactorApp.Program2*
Editor Edit View Insert Tools Window Help
Check (Ctrl+B)
Variables
Function Blocks
Code
Description
Check
Find in Editor

	Nam	Data Type	Attributes	Initial Value	I/O Address	Access Variab
1	SW1	BoolIO			Controller_1.	
2	SW2	BoolIO			Controller_1.	
3	SW3	BoolIO			Controller_1.	
4	lamp1	BoolIO				
5						

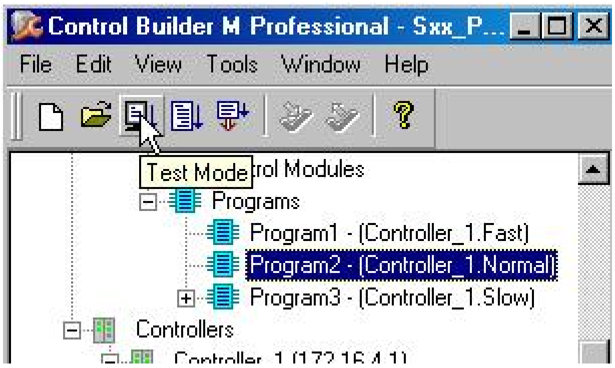
```
lamp1.Value := (SW1.Value AND SW2.Value) OR SW3.Value;
```

```
Starting check 2006-08-18-16:00:28.
Summary: Time 1 second, 0 Errors, 0 warnings.
```

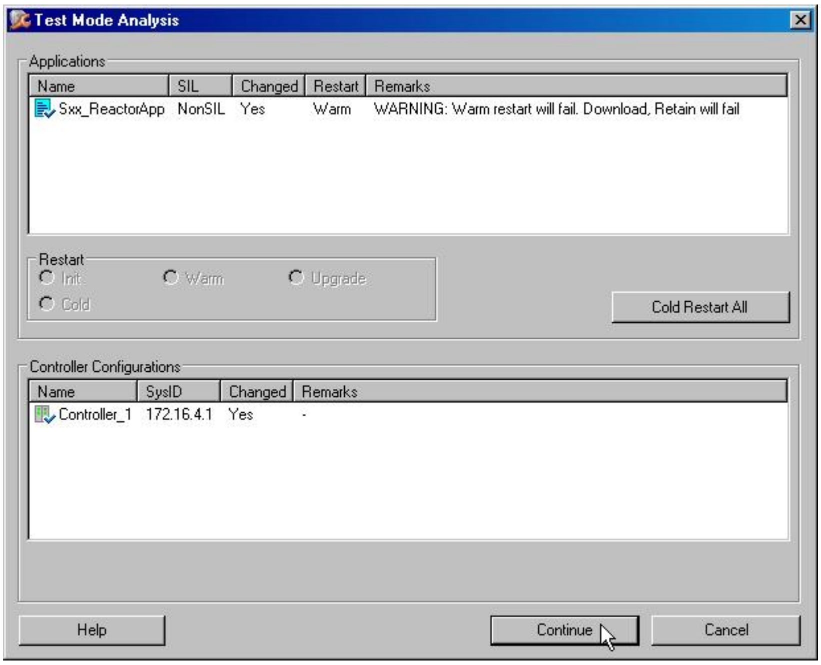
Checks the contents of the editor   Row 1, Col 34   800xAadmin

5. Save and close the POU editor.

6. Select “Test Mode”.


Control Builder M Professional - Sxx_P...
File Edit View Tools Window Help
Test Mode
rol Modules
Programs
Program1 - (Controller_1.Fast)
Program2 - (Controller_1.Normal)
Program3 - (Controller_1.Slow)
Controllers
Controller_1 (172.16.4.1)

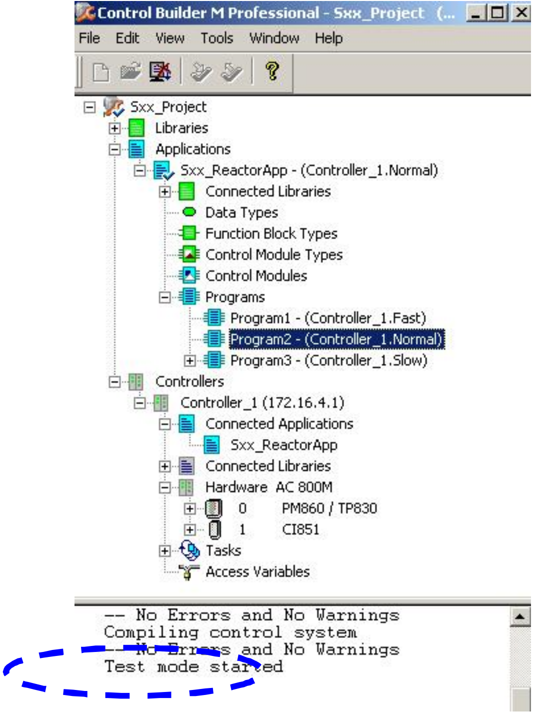
7. Click the “Continue” button for the On Line Change analysis dialog and observe that Control Builder enters the “Test” mode



Name	SIL	Changed	Restart	Remarks
Sxx_ReactorApp	NonSIL	Yes	Warm	WARNING: Warm restart will fail. Download, Retain will fail

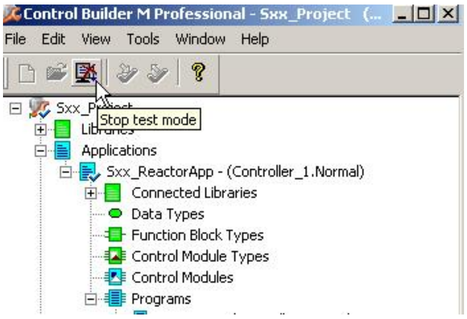
Name	SysID	Changed	Remarks
Controller_1	172.16.4.1	Yes	-

8. Check the Message Pane in Control Builder.



```
-- No Errors and No Warnings
Compiling control system
-- No Errors and No Warnings
Test mode started
```

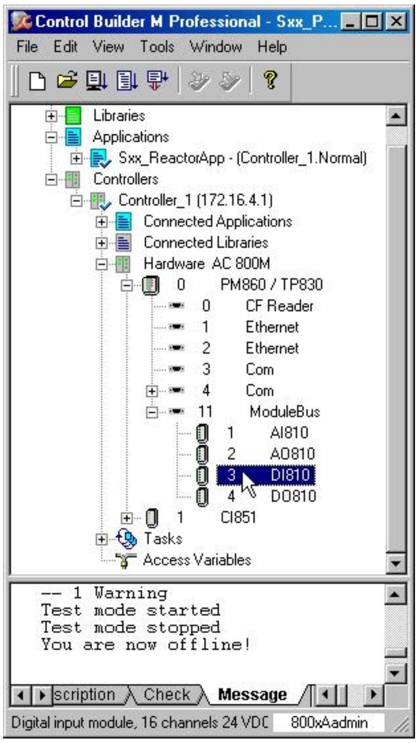
9. Take Control Builder to the Off-line mode by pressing the “Stop Test Mode” icon on the toolbar.



6.2.3.3 Connect Variables to IO Channels

☞ Connect the three switches on the DI810 module to the SW1, SW2, and SW3 variable you created in Program2

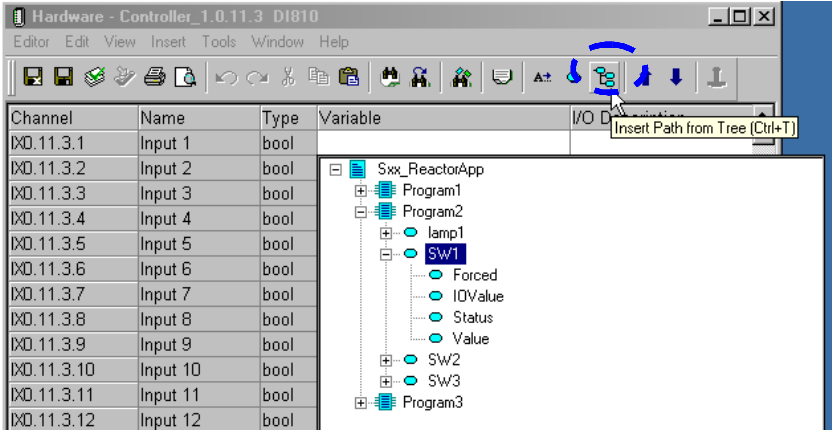
1. Open up the hardware structure in the Control Builder and double click on the DI810 module in the tree:



```
-- 1 Warning
Test mode started
Test mode stopped
You are now offline!
```

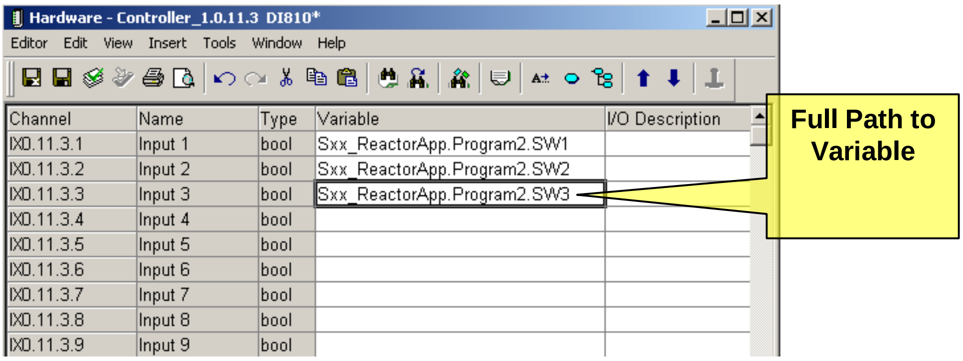
Digital input module, 16 channels 24 VDC 800xAadmin

2. Connect the three switches on the DI810 module to the *SW1*, *SW2*, and *SW3* variables you created in “Program2”. Use the icons in the upper right corner of the editor to choose variables. Open the tree to view the variables, select the +


Hardware - Controller_1.0.11.3 DI810
Editor Edit View Insert Tools Window Help
Insert Path from Tree (Ctrl+T)
Sxx_ReactorApp
Program1
Program2
lamp1
SW1
Forced
IOValue
Status
Value
SW2
SW3
Program3

Channel	Name	Type	Variable	I/O Description
IX0.11.3.1	Input 1	bool		
IX0.11.3.2	Input 2	bool		
IX0.11.3.3	Input 3	bool		
IX0.11.3.4	Input 4	bool		
IX0.11.3.5	Input 5	bool		
IX0.11.3.6	Input 6	bool		
IX0.11.3.7	Input 7	bool		
IX0.11.3.8	Input 8	bool		
IX0.11.3.9	Input 9	bool		
IX0.11.3.10	Input 10	bool		
IX0.11.3.11	Input 11	bool		
IX0.11.3.12	Input 12	bool		

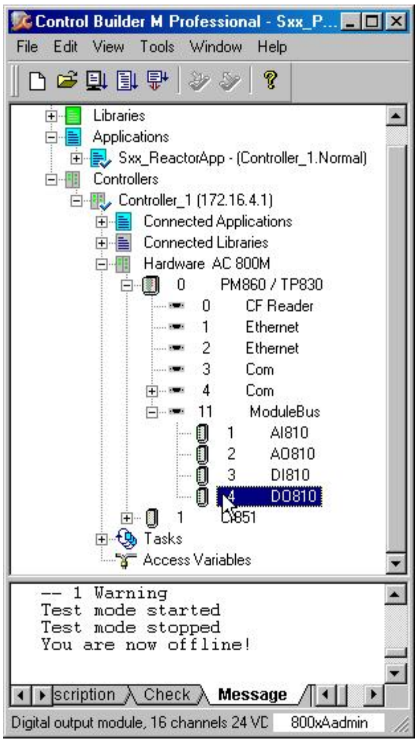
3. Save and close the editor when finished.


Hardware - Controller_1.0.11.3 DI810*
Editor Edit View Insert Tools Window Help

Channel	Name	Type	Variable	I/O Description
IX0.11.3.1	Input 1	bool	Sxx_ReactorApp.Program2.SW1	
IX0.11.3.2	Input 2	bool	Sxx_ReactorApp.Program2.SW2	
IX0.11.3.3	Input 3	bool	Sxx_ReactorApp.Program2.SW3	
IX0.11.3.4	Input 4	bool		
IX0.11.3.5	Input 5	bool		
IX0.11.3.6	Input 6	bool		
IX0.11.3.7	Input 7	bool		
IX0.11.3.8	Input 8	bool		
IX0.11.3.9	Input 9	bool		

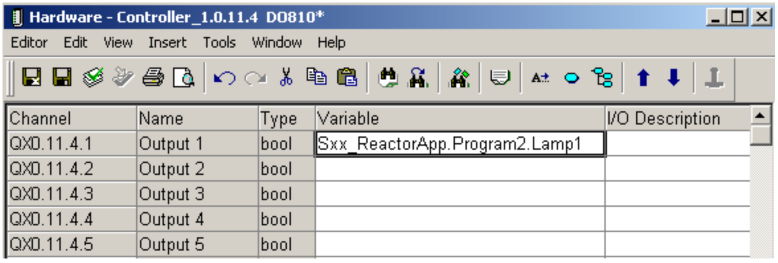
Full Path to Variable

4. Select DO810



```
-- 1 Warning
Test mode started
Test mode stopped
You are now offline!
```

5. Connect the first channel on the DO810 module to the *Lamp1* variable you created in *Program2*. Check for errors and then save and close the editor when finished.


Hardware - Controller_1.0.11.4 DO810*
Editor Edit View Insert Tools Window Help

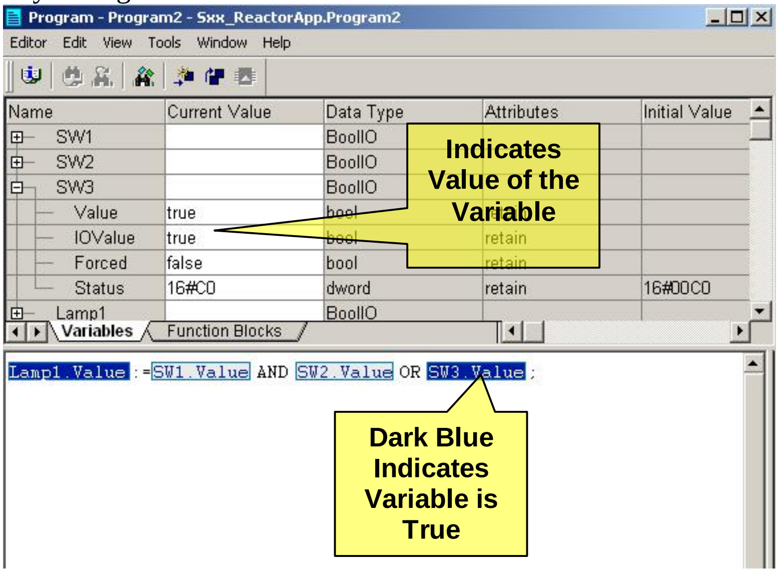
Channel	Name	Type	Variable	I/O Description
QX0.11.4.1	Output 1	bool	Sxx_ReactorApp.Program2.Lamp1	
QX0.11.4.2	Output 2	bool		
QX0.11.4.3	Output 3	bool		
QX0.11.4.4	Output 4	bool		
QX0.11.4.5	Output 5	bool		

6. Download and Go Online. Select “Continue”

6.2.3.4 View Code in Online Mode

Check Control Builder is in the On-Line mode and open the POU editor for Program 2 and observe the logic is in on-line mode.

1. Manipulate the three switches and observe the light on the IO panel while looking at your logic.


Program - Program2 - Sxx_ReactorApp.Program2
Editor Edit View Tools Window Help
Variables
Function Blocks

Name	Current Value	Data Type	Attributes	Initial Value
SW1		BoolIO		
SW2		BoolIO		
SW3		BoolIO		
Value	true	bool		
IOValue	true	bool	retain	
Forced	false	bool	retain	
Status	16#C0	dword	retain	16#00C0
Lamp1		BoolIO		

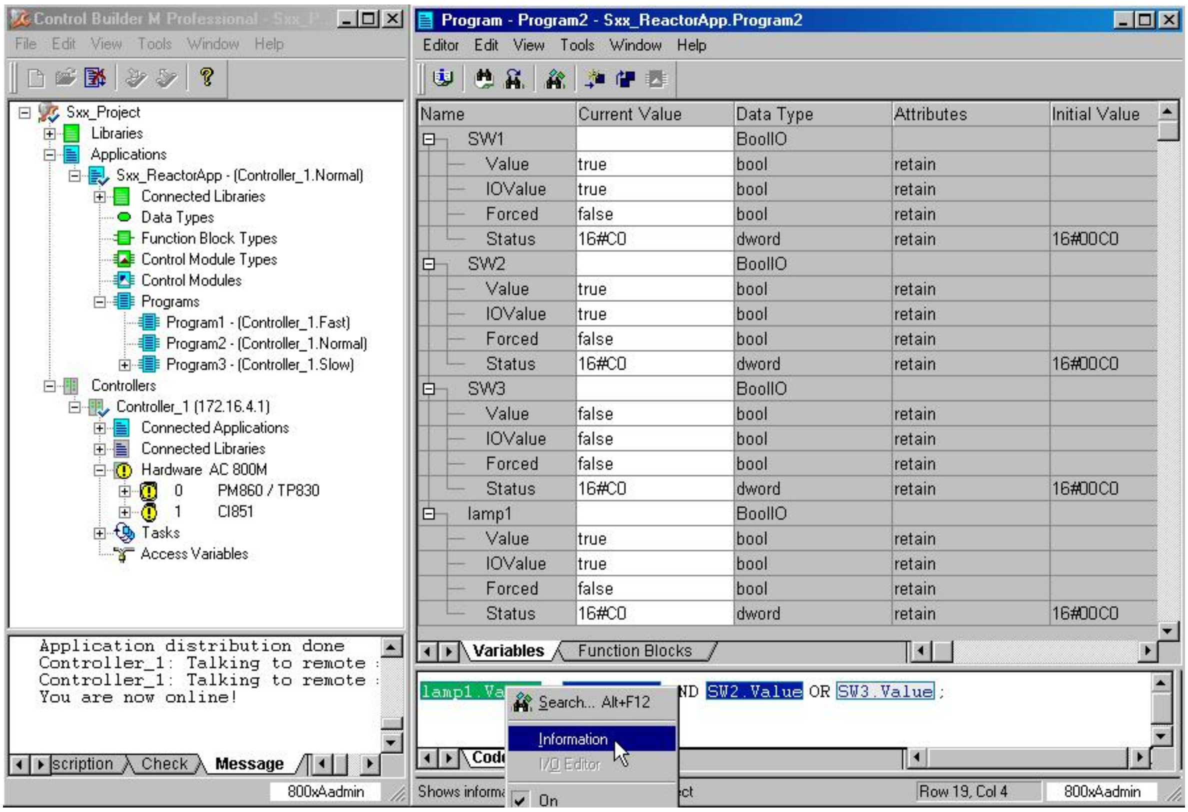
```
Lamp1.Value :=SW1.Value AND SW2.Value OR SW3.Value ;
```

Indicates Value of the Variable

Dark Blue Indicates Variable is True

2. .
   Right click on the variable in the Declaration Pane to get information about the variable and also to open the I/O editor. on

3. For information right click at a variable

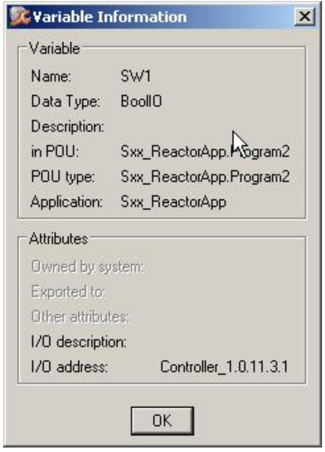


```
Application distribution done
Controller_1: Talking to remote
Controller_1: Talking to remote
You are now online!
```

Name	Current Value	Data Type	Attributes	Initial Value
SW1		BoolIO		
Value	true	bool	retain	
IOValue	true	bool	retain	
Forced	false	bool	retain	
Status	16#C0	dword	retain	16#00C0
SW2		BoolIO		
Value	true	bool	retain	
IOValue	true	bool	retain	
Forced	false	bool	retain	
Status	16#C0	dword	retain	16#00C0
SW3		BoolIO		
Value	false	bool	retain	
IOValue	false	bool	retain	
Forced	false	bool	retain	
Status	16#C0	dword	retain	16#00C0
lamp1		BoolIO		
Value	true	bool	retain	
IOValue	true	bool	retain	
Forced	false	bool	retain	
Status	16#C0	dword	retain	16#00C0

```
lamp1.Va        ND SW2.Value OR SW3.Value;
```

4. Read the information regarding to I/O address and Application



Variable

Name: SW1

Data Type: BoolIO

Description:

in POU: Sxx_ReactorApp.Program2

POU type: Sxx_ReactorApp.Program2

Application: Sxx_ReactorApp

Attributes

Owned by system:

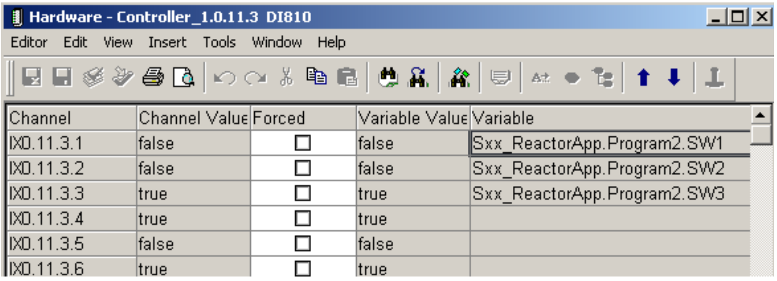
Exported to:

Other attributes:

I/O description:

I/O address: Controller_1.0.11.3.1

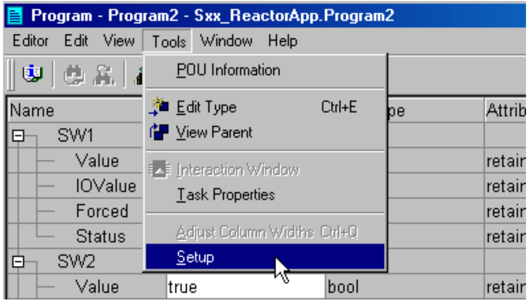
5. Select the hardware tab “Status” to view the connections


Hardware - Controller_1.0.11.3 DI810
Editor Edit View Insert Tools Window Help

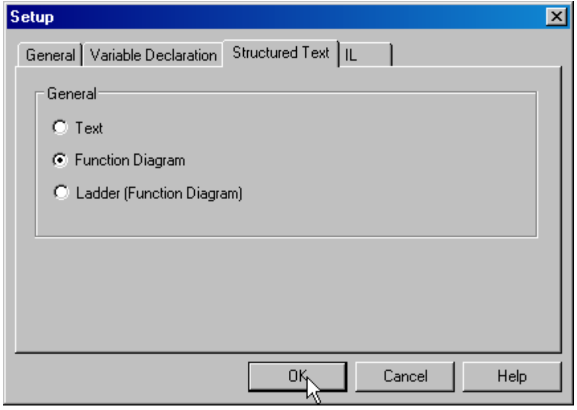
Channel	Channel Value	Forced	Variable Value	Variable
IX0.11.3.1	false	□	false	Sxx_ReactorApp.Program2.SW1
IX0.11.3.2	false	□	false	Sxx_ReactorApp.Program2.SW2
IX0.11.3.3	true	□	true	Sxx_ReactorApp.Program2.SW3
IX0.11.3.4	true	□	true	
IX0.11.3.5	false	□	false	
IX0.11.3.6	true	□	true	

6.2.3.5 Change the Online View of Structured Text (ST)

1. Select **Tools** > **Setup** from the program editor menu bar.



2. Select radio button for Function Diagram, Select “OK”



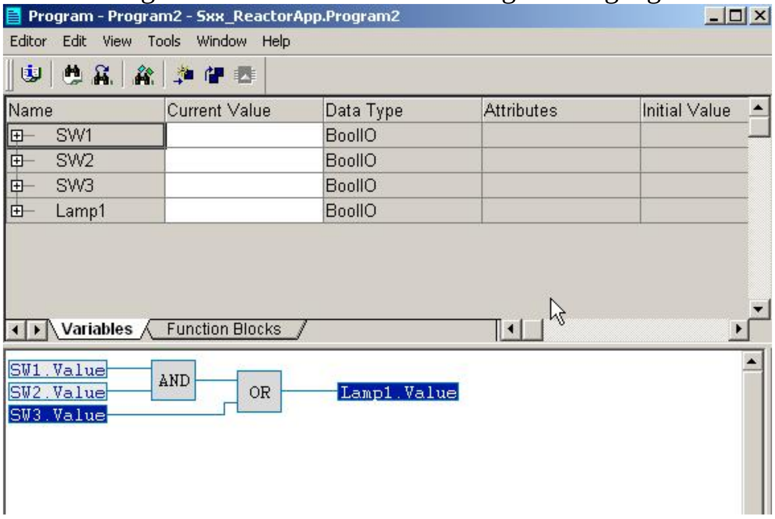
General

○ Text

● Function Diagram

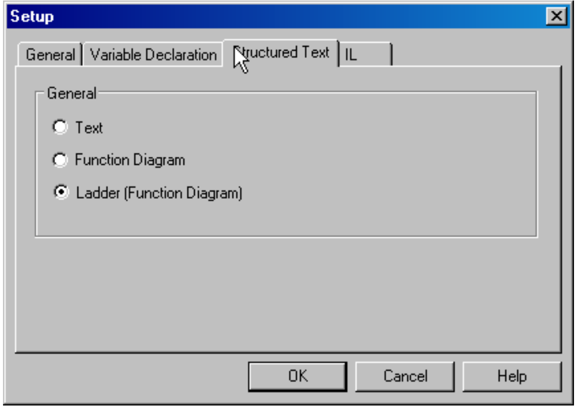
○ Ladder (Function Diagram)

3. View the logic on-line in the Function Diagram language.


Program - Program2 - Sxx_ReactorApp.Program2
Editor Edit View Tools Window Help
Variables
Function Blocks
SW1.Value
SW2.Value
SW3.Value
AND
OR
Lamp1.Value

Name	Current Value	Data Type	Attributes	Initial Value
SW1		BoolIO		
SW2		BoolIO		
SW3		BoolIO		
Lamp1		BoolIO		

4. View the logic on-line in the Ladder language.



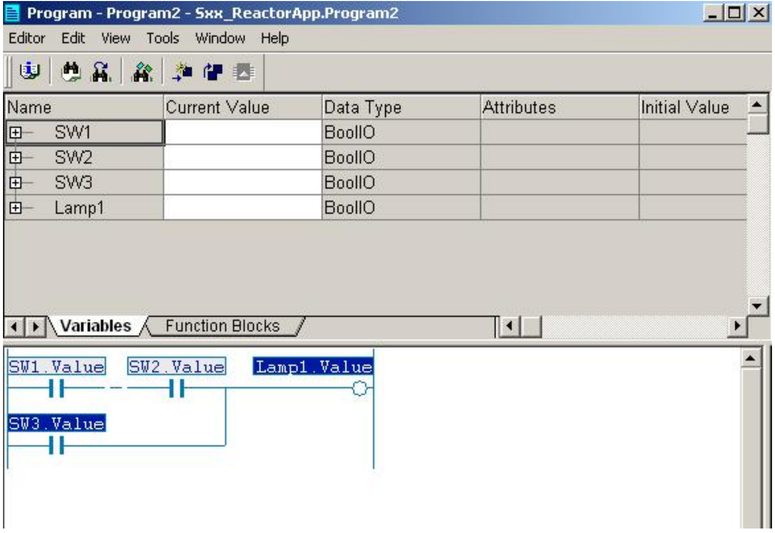
General

○ Text

○ Function Diagram

● Ladder (Function Diagram)

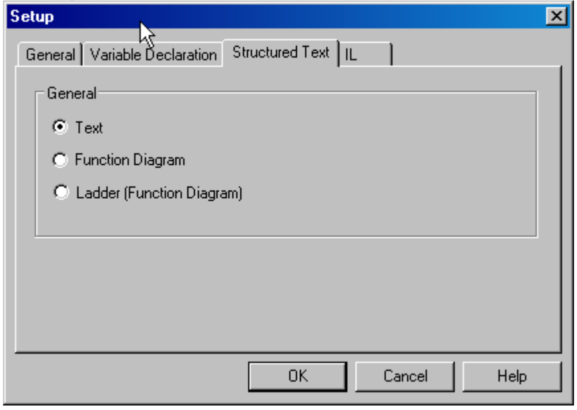
5. Change a switch at the panel.



Program - Program2 - Sxx_ReactorApp.Program2
Editor Edit View Tools Window Help
Variables
Function Blocks
SW1.Value
SW2.Value
Lamp1.Value
SW3.Value

Name	Current Value	Data Type	Attributes	Initial Value
SW1		BoolIO		
SW2		BoolIO		
SW3		BoolIO		
Lamp1		BoolIO		

6. Change the view back to Text.



General

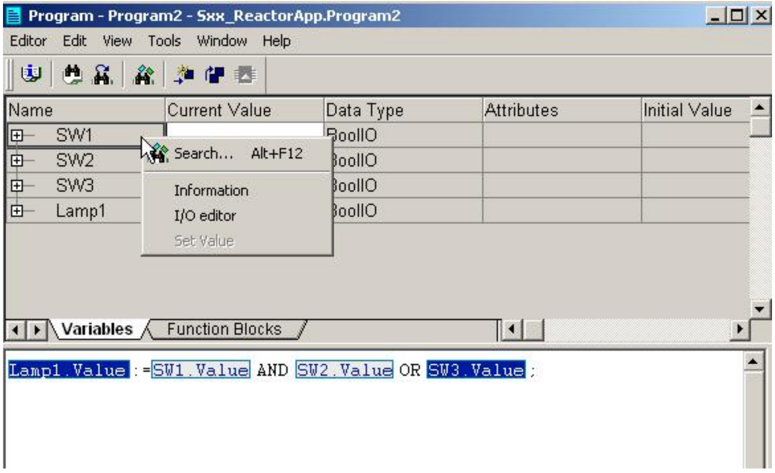
● Text

○ Function Diagram

○ Ladder (Function Diagram)

☞ Forcing IO

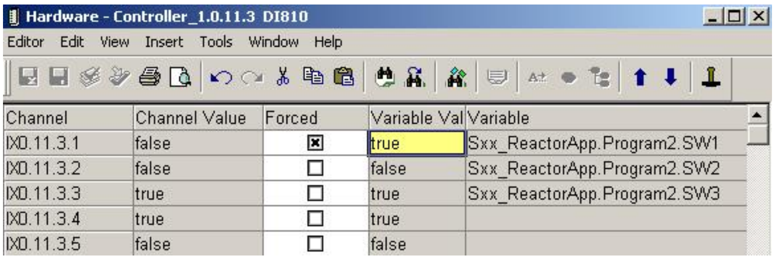
1. Open the hardware editor screen for the DI810 module by right-clicking the *SW1* variable in the program editor and selecting “I/O Editor” from the context menu.


Program - Program2 - Sxx_ReactorApp.Program2
Editor Edit View Tools Window Help
Search... Alt+F12
Information
I/O editor
Set Value
Variables
Function Blocks

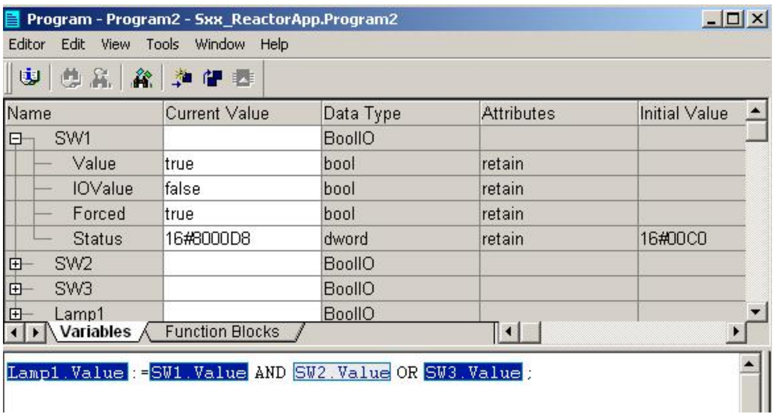
Name	Current Value	Data Type	Attributes	Initial Value
⊞ SW1		BoolIO		
⊞ SW2		BoolIO		
⊞ SW3		BoolIO		
⊞ Lamp1		BoolIO		

```
Lamp1.Value :=SW1.Value AND SW2.Value OR SW3.Value ;
```

2. Select the “Status”, and force one or more variables. Observe the resulting changes in your Program2 editor’s code pane as you force and release forces on variables.


Hardware - Controller_1.0.11.3 DI810
Editor Edit View Insert Tools Window Help

Channel	Channel Value	Forced	Variable Val	Variable
IX0.11.3.1	false	☒	true	Sxx_ReactorApp.Program2.SW1
IX0.11.3.2	false	□	false	Sxx_ReactorApp.Program2.SW2
IX0.11.3.3	true	□	true	Sxx_ReactorApp.Program2.SW3
IX0.11.3.4	true	□	true	
IX0.11.3.5	false	□	false	


Program - Program2 - Sxx_ReactorApp.Program2
Editor Edit View Tools Window Help
Variables
Function Blocks

Name	Current Value	Data Type	Attributes	Initial Value
⊟ SW1		BoolIO		
Value	true	bool	retain	
IOValue	false	bool	retain	
Forced	true	bool	retain	
Status	16#8000D8	dword	retain	16#00C0
⊞ SW2		BoolIO		
⊞ SW3		BoolIO		
⊞ Lamp1		BoolIO		

```
Lamp1.Value :=SW1.Value AND SW2.Value OR SW3.Value ;
```