Topic: Sequential Control Module - Lab

Contents

Instructions for Sequential Control Module - Lab .................................................................3
Add Function Blocks for SCM Program Control .....................................................................5
Specify SCM Recipe Values and Invoke Transition ..............................................................7
Configure an SCM Abort Sequence .........................................................................................21
Use a Step to Start Another SCM ..........................................................................................27
Use Recipe Values to Set Minimum Flow ...............................................................................33
Interface an SCM with a Graphic Textbox ............................................................................37
R410: Use DATA Block (Formula Parameter) to set minimum Flow ........................................43
This page was intentionally left blank.
Instructions for Sequential Control Module - Lab

Introduction

This portion of the document contains important information required to perform to ‘SCM Lab’ lab.

Procedure

<table>
<thead>
<tr>
<th>✓</th>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>Read the following IMPORTANT information carefully.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Decide if you will be performing labs for C200E (and Series A I/O modules) OR C300 (and Series C I/O Modules). You can opt for either one but not both.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>______ C200E (and Series A I/O modules)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>______ C300 (and Series C I/O Modules)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>ATTENTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You must perform the remaining steps of this lab if you skipped any of the prior lessons of this course, else skip the remaining instructions and proceed to the next lab.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Locate the “GUI_DB_Load” Shortcut on desktop.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Double click “GUI_DB_Load” Shortcut to launch the import utility.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. In Controller drop down select either C200 or C300 Controller (as per the option selected in step#1).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. In Lesson number drop down select Lesson23.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. Click the LOAD button.</td>
</tr>
</tbody>
</table>
### Instructions for Sequential Control Module - Lab

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td><strong>ATTENTION</strong>&lt;br&gt;If you are facing any of the following situation while loading the data base:&lt;br&gt;1. Import wrong Lesson Database&lt;br&gt;2. Error prompt appears while loading data base.&lt;br&gt;Then students should wait for at least one minute before reloading the database.</td>
</tr>
<tr>
<td>4</td>
<td>Open Control Builder. Select ONLY the Controller and load it. Activate the CEE using <strong>Warmstart</strong> option.</td>
</tr>
<tr>
<td>5</td>
<td>Now select and load all IO and Control Modules with the “<strong>Automatically change All Control ..... .....</strong>” option selected.</td>
</tr>
<tr>
<td>6</td>
<td>Ensure that the Controller, IO, and Control Modules are indicated in green color.</td>
</tr>
<tr>
<td>7</td>
<td>Open the Excel Spreadsheet if not already open. Open either from <strong>C:\Users\Student\Documents\Tie_Back_New.xls OR</strong> by double clicking the shortcut on your desktop. Proceed to next Lab</td>
</tr>
</tbody>
</table>
Add Function Blocks for SCM Program Control

Objective

- Add five flag function blocks and one numeric function block to the existing Flags CM

Prerequisites

- Experion PKS Server with all required CMs loaded
- D_100.htm graphic loaded on the server. (This is required at a later time to check the control strategy.)
- Control Builder running with one or two Project/Monitor tree windows open
- Excel Data Exchange open with the simulation spread sheet loaded

Introduction

In this lab, you will add five flag function blocks and one numeric function block to the existing FLAGS CM for later use in the SCM program labs.

NOTE: You will be given the following:

<table>
<thead>
<tr>
<th>BLOCK NAME</th>
<th>Action to be taken</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLAGS</td>
<td>Add five Flag Function Blocks and one Numeric Function Blocks to the existing Flags CM.</td>
</tr>
</tbody>
</table>
Procedure

Add Function Blocks to FLAGS

<table>
<thead>
<tr>
<th>✓</th>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>Open the Chart view of the CM called FLAGS in project window.</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Add five Flags and one Numeric to the existing FLAGS CM. Name the new flags P_START, F_MESSAGE, READY, P_ABORT and PROCESS. Name the numeric FLOW. Save and close FLAGS. Load and activate FLAGS.</td>
</tr>
</tbody>
</table>
Specify SCM Recipe Values and Invoke Transition

Objective

- Configure recipe values for later use in setting minimum flow level
- Configure the invoke transition
- Configure a step to turn on the Warning message
- Use a transition to check the Ready flag
- Configure a verify step, for later use, to allow the SCM to take separate paths
- Configure a Sync to allow the SCM to follow parallel paths
- Check and verify SCM program operation to this point

Prerequisites

- Experion PKS Server with all required CMs built and loaded
- D_100.htm graphic loaded on the server (This is required at a later time to check the control strategy.)
- Control Builder running with one or two Project/Monitor tree windows open
- Excel Data Exchange open with simulation spread sheet loaded
- Previous SCM lab in this section complete, tested, and verified

Introduction

The process described in this lab is totally fictitious and created only to demonstrate SCM features.

In the first portion of this SCM program you will:

- Write a sequence that will do a fan switch check.
- Write and verify that the SCM will follow either of two routes based on the fan switch selection
- Use a set of Sync blocks to set up parallel sequence paths
- **NOTE:** You will configure the following:

<table>
<thead>
<tr>
<th>BLOCK NAME</th>
<th>Action to be taken</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCM_D100</td>
<td>Write a step sequence to start circulation, and then initiate full process flow.</td>
</tr>
</tbody>
</table>
Create a SCM

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | a. Build a new SCM into your project.  
      b. Open the SCM in project chart view.  
      c. Open the configuration parameters window for this SCM.  
      d. Enter the following details on the Main tab:  
        Name: SCM_D100  
        Item Name: SCM_D100_Item  
        Description: Circulate / Process  
        Parent Asset: C11 |

<table>
<thead>
<tr>
<th>Index</th>
<th>Parameter Descr</th>
<th>Target Value</th>
<th>Target Hi</th>
<th>Target Lo</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CIRCULATE_LOW_FLOW</td>
<td>47</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>CIRCULATE_LOW_FLOW_2</td>
<td>850</td>
<td>1750</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>CIRCULATE_LOW_FLOW_3</td>
<td>1</td>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>

Accept the defaults for any other remaining parameters.  
Click OK.

2 On the Recipe Tab, right-click in the blank recipe area and select Append Row).  
Add three rows (recipe values) and configure them as follows:

<table>
<thead>
<tr>
<th>Index</th>
<th>Parameter Descr</th>
<th>Target Value</th>
<th>Target Hi</th>
<th>Target Lo</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CIRCULATE_LOW_FLOW</td>
<td>47</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>CIRCULATE_LOW_FLOW_2</td>
<td>850</td>
<td>1750</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>CIRCULATE_LOW_FLOW_3</td>
<td>1</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>✓ Step</td>
<td>Action</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 3 | a. Add one condition to the Invoke Transition.  
   b. Open the Transition for edit.  
   Enter the following:  
   Transition Name:  
   **Start_Check**  
   Transition Description:  
   **Start Check**  
   Condition 1 description:  
   **Check Start Flag**  
   Condition 1:  
   $\text{FLAGS.P\_START.PVFL} = 1$  
   Primary Gate:  
   **Connect**  
   Secondary Gate:  
   **Connect**  
   
   Click **OK**. |

**Condition**

<table>
<thead>
<tr>
<th>#</th>
<th>Transition Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\text{FLAGS.P_START.PVFL} = 1$</td>
</tr>
</tbody>
</table>

**Description**

<table>
<thead>
<tr>
<th>#</th>
<th>Transition Condition Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Check Start Flag</strong></td>
</tr>
</tbody>
</table>
## Step 4

**Action**

- Drag a STEP from the library into your project.
- Add two Step outputs.

Enter the following:

<table>
<thead>
<tr>
<th>Step Name</th>
<th>M_ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Description</td>
<td>Message On</td>
</tr>
<tr>
<td>Output 1 Description</td>
<td>ABORT FLAG OFF</td>
</tr>
</tbody>
</table>

**Output 1 Expression**

```
FLAGS.P_ABORT.PVFL := 0
```

<table>
<thead>
<tr>
<th>Output 2 Description</th>
<th>START MESSAGE ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output 2 Expression</td>
<td><code>FLAGS.F_MESSAGE.PVFL := 1</code></td>
</tr>
</tbody>
</table>

Click **OK**.

Wire the **Transition** above to this **Step**.

### ATTENTION

From now on wire the Steps and Transitions, as needed.

#### Output Expressions

<table>
<thead>
<tr>
<th>#</th>
<th>Step Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>FLAGS.P_ABORT.PVFL := 0</code></td>
</tr>
<tr>
<td>2</td>
<td><code>FLAGS.F_MESSAGE.PVFL := 1</code></td>
</tr>
</tbody>
</table>

#### Output Descriptions

<table>
<thead>
<tr>
<th>#</th>
<th>Step Output Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Abort Flag OFF</td>
</tr>
<tr>
<td>2</td>
<td>Start Message ON</td>
</tr>
</tbody>
</table>
5

a. Drag a Transition from the library into your project.

b. Enter the following:

   Transition Name:
   **Check Ready**

   Transition Description:
   **Check Ready**

   Condition:
   Create an expression that checks to see if **FLAGS.READY** is on.
   Hint: See The Start_Check Transition in an earlier step for an example expression.

   Description:
   **Check Ready Flag**

   Primary Gate:
   **Connect**

   Secondary Gate:
   **Connect**
Sequential Control Module - Lab
Specify SCM Recipe Values and Invoke Transition

<table>
<thead>
<tr>
<th>✓ Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 6     | a. Drag a new STEP into your project.  
b. Enter the following:  
Step Name: VERIFY  
Step Description: VERIFY  

ATTENTION  
Use this step only as a decision point. No expression is needed. One branch will continue to follow the process and the other branch will lead to an abort sequence. |

| 7     | a. Select the Block Pins Tab.  
c. Select the Block Preferences tab.  
d. Check the View Pin Labels option.  
e. Click OK. |

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Label</th>
<th>Type</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEXTCOMP[1]</td>
<td>NO</td>
<td>YES</td>
<td>OUTPUT</td>
<td>BOTTOM</td>
</tr>
<tr>
<td>NEXTCOMP[2]</td>
<td>NO</td>
<td>YES</td>
<td>OUTPUT</td>
<td>BOTTOM</td>
</tr>
<tr>
<td>LESC</td>
<td>NO</td>
<td>YES</td>
<td>INPUT</td>
<td>TOP</td>
</tr>
<tr>
<td>Step</td>
<td>Action</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>a. Drag a Sync block into your project under the VERIFY step.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>b. Wire from VERIFY NEXTCOMP[1] pin to the input of the Sync block.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>c. Leave all parameters on the Sync block as default.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Diagram showing the connection of VERIFY and SYNCA blocks with wires from and to NEXTCOMP[1] and NEXTCOMP[2] pins.](image-url)
<table>
<thead>
<tr>
<th>✓ Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 9     | a. Drag two new Transitions into your project. Arrange them side by side under the Sync block. *(See Step 13.)*  
|       | b. Add four Expressions to each transition.  
|       | c. Enter the following in the left transition:  
|       | **Transition Name:** CHECK_FANS  
|       | **Transition Description:** CHECK_FANS  
|       | Create four conditions as described below:  
|       | **Condition Description:** CHECK HS14A((through D).FLAGA = ON) (Add similar descriptions to all four expressions).  
|       | **Condition:** Create four expressions that check to see if 11_HS14A((through D).FLAGA.PVFL = ON (=1).  
|       | **Primary Gate:** AND (all four expressions to P1)  
|       | **Secondary Gate:** Connect  
|       | Click OK.  

**Diagram:**

```
<table>
<thead>
<tr>
<th>#</th>
<th>Transition Condition Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CHECK HS14A.FLAGA = ON</td>
</tr>
<tr>
<td>2</td>
<td>CHECK HS14B.FLAGA = ON</td>
</tr>
<tr>
<td>3</td>
<td>CHECK HS14C.FLAGA = ON</td>
</tr>
<tr>
<td>4</td>
<td>CHECK HS14D.FLAGA = ON</td>
</tr>
</tbody>
</table>
```

Click **OK**.
### Step 10

Enter the following in the right transition:

**Transition Name:**

`CHECK_FANS2`

**Transition Description:**

`CHECK_FANS2`

Create four conditions as described below:

**Condition Description:**

`CHECK HS14E` (through `H`).`FLAGA = ON` (Add similar descriptions to all four expressions).

**Condition:**

Create four expressions that check to see if

`11_HS14E` (through `H`).`FLAGA.PVFL = ON` (=1)

**Primary Gate:**

`AND` (all four expressions to `P1`)

**Secondary Gate:**

Connect

Click **OK**.

### Step 11

- **a)** Add a step under each of the transitions.
- **b)** Add one output to each step.
- **c)** Enter the following in the **left** step:
  - **Step Name:** `M_OFF`
  - **Step Description:** `MESSAGE OFF`
  - **Output 1 Expression:** Write an expression that will turn `Flags.F_MESSAGE OFF`
  - **Output 1 Description:** Warning message flag off

- **d)** Enter the following in the **right** step:
  - **Step Name:** `S_FLAG_O`
  - **Step Description:** `START FLAG OFF`
  - **Output 1 Expression:** Write an expression that will turn `FLAGS.P_START OFF`
  - **Output 1 Description:** Start Flag Off
<table>
<thead>
<tr>
<th>✓ Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| **12** | a. Drag another Sync block into your project under the two Steps.  
b. Make the name of this Sync block: **SYNC_1**  
c. Remove NEXTCOMP[2] pin from the output of this Sync block.  
d. Leave all parameters on the Sync block as default. |
| **13** | Wire the function blocks together, as shown below. |
Step 14

At this point, your SCM should look similar to the following:

**TIP**

For easier branch viewing, zoom out to 50%.
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 15   | **Check the program.**  
|      | a. Save the SCM, assign it to your CEE (CEESCEFB61 or SIM_CEEC300) and load **SCM_D100**.  
|      | b. Select the monitoring tab and make **SCM_D100** active.  
|      | c. Make sure all CMs are loaded and active.  
|      | d. Verify the simulation spreadsheet is running.  
|      | e. In Station, call up the graphic **D_100**. |

| 16   | Click on SCM_D100 button. Change the **CIRCULATE** combo box to **ON** (this sets Flags_P_Start.PV = ON. This also causes PVFL to be ON which makes the condition for the invoke transition of SCM_D100 to be true).  
|      | At the bottom of the display you should see that the Active Handler is **SCM_D100.Main** and the Active Step is **SCM_D100.M_ON**.  
|      | You should also see the Warning message in the lower right corner of the display.  
|      | Click the **Fan Control** button.  
|      | You should now see the Fan Control Panel pop up display.  
|      | Leave any one fan switch in the **OFF** position and turn the rest of the fan switches to **ON** position. |

![Diagram of Fan Control Panel](image-url)
17 Select the **SAFE** combo box and change it to **ON** (this sets Flags.Ready.PV = ON. This also causes PVFL to be ON).
The Program Active Step should now be **SCM_D100.VERIFY**.

18 Open Control Builder and view **SCM_D100** in Monitoring mode and perform the following:
   - Verify that there are no red Steps or Transitions.
   - Verify that all the expressions in the **CHECK_FANS** and **CHECK_FANS2** Transitions are green, except for the one that matches the fan switch that was left in the **OFF** position.

19 Turn the other fan on and verify that the SCM completes.
<table>
<thead>
<tr>
<th>✓ Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| **20** | If necessary to re-enforce understanding, run the SCM several more times. To prepare the system, and run the SCM again:  
- From Station, on the D_100 display, turn both the **CIRCULATE** and the **SAFE** flags  **OFF**.  
- In Control Builder, change the state of **SCM_D100** to **IDLE**.  
- Change the Mode Attribute of **SCM_D100** to **Operator** and change the state to **IDLE**.  
- Run the program  
Run the SCM with at least one fan switch off.  
The expected result is that the SCM stops at the **VERIFY** Step. (Only the first 3 function blocks will be blue.)  
Run the SCM with all fan switches on.  
The expected result is that the SCM will complete execution through the last Sync block. (Eight function blocks will be blue including the parallel transitions and steps.)  
Fix any encountered problems. |
| **21** | When satisfied with your program:  
Call up the **D_100** graphic in Station and set the **CIRCULATE** and **SAFE** combo boxes to **OFF**.  
In Control Builder, change the state of **SCM_D100** to **Idle**, and **inactivate** it.  
Change the Mode Attribute of **SCM_D100** to **Operator** and change the state to **IDLE**. |
Configure an SCM Abort Sequence

Objective

- Configure an abort sequence

Prerequisites

- Experion PKS Server with all required CMs loaded
- D_100.htm graphic loaded on the Server. (This is required at a later time to check the control strategy.)
- Control Builder running with one or two Project/Monitor tree windows open
- Excel Data Exchange open with the simulation spread sheet loaded
- All previous SCM labs in this section completed, tested, and verified

Introduction

This part of the program will execute an abort sequence.

The process described in this lab is totally fictitious and created only to demonstrate SCM features.

- In this portion of the SCM, you will add programming to turn off the Warning message and turn on an Abort Message
Build an SCM Abort Sequence

<table>
<thead>
<tr>
<th>✓ Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1      | The steps in this lab will have you:  
Add a set of steps and transitions to begin a safety shut down sequence. In a later module, you will complete this action by adding an **Abort** Handler to this branch of the SCM.  
Add the Abort steps and transitions to the right of the main program.  
Add other steps and transitions (not the abort steps and transitions) under the **SYNC_1** block. |
<table>
<thead>
<tr>
<th>✓ Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>2</strong></td>
<td>Open SCM_D100 in project view.</td>
</tr>
</tbody>
</table>

Drag a transition from the library into your project.  
Add a condition to the transition.  
Enter the following:  
Transition Name:  
**DUMMY**  
Transition Description:  
**DUMMY**  
Condition Description:  
**DUMMY**  
Condition Expression:  
1=1  
Primary Gate:  
**Connect**  
Secondary Gate:  
**Connect**  
You must add a transition here because the sequence branched at a step. In this case, there was no useful action needed by this transition, so a dummy was used.

<p>| <strong>3</strong>  | Connect and wire the NEXTCOMP[2] pin on the ‘Verify’ step to the Dummy transition you just added – as shown below: |</p>
<table>
<thead>
<tr>
<th>✓ Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Drag a STEP from the library into your project and place it under your Dummy transition. Add two Step outputs. Enter the following: Step Name: <strong>M_OFF_FAIL_ON</strong> Step Description: <strong>M_OFF_FAIL_ON</strong> Min Wait Time: 10</td>
</tr>
</tbody>
</table>

**ATTENTION**

Min Wait Times and Max Active Times are given in execution cycles. In this case, if the SCM scan time is one second, then Min Wait Time would be 10 seconds. You may wish to adjust this parameter to a longer time during the check out procedure.

Max Active Time: 240

Output 1 Description: **TURN WARNING MESSAGE OFF**
Output 1 Expression: Create an expression that will turn **FLAGS.F_MESSAGE** OFF.

Output 2 Description: **TURN ABORT MESSAGE ON**
Output 2 Expression: Create an expression that will turn **FLAGS.P_ABORT** ON.
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 5    | **Add another dummy transition below the M_OFF_FAIL_ON Step.**  
Transition Name:  
**DUMMY2**  
Transition Description:  
**DUMMY2** |
| 6    | **Drag a STEP from the library into your project.**  
Add one Step output.  
Enter the following:  
Step Name:  
**ABORT_OFF**  
Step Description:  
**ABORT_OFF**  
Output 1 Description:  
**Turn Abort Message OFF**  
Output 1 Expression:  
Create an expression to turn off **FLAGS.P_ABORT**. |
| 7    | **Program check:**  
Complete the wiring of the steps and transitions.  
Save, close and load SCM_D100.  
Select the monitoring tab and make SCM_D100 active.  
Make sure all the other CMs are loaded and active.  
Check the FLAGS CM and make sure all the contained flag function blocks are off.  
In Station call up the graphic D_100. |
<table>
<thead>
<tr>
<th>✓ Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 8      | Run the SCM as you did in the previous program check. Verify the following:  
- When all fan switches are ON, the SCM executes the “normal” path (through SYNCA).  
- When one or more of the fan switches are OFF, the SCM executes the “abort” path (we just added).  
After the sequence runs with at least one fan switch in the OFF position, the message will indicate **FAN FAILURE** and **PROGRAM ABORT**. The Active Step will also be blank.  

**ATTENTION**  
The Fan Failure message will disappear after a few seconds. The Active Step will also be blank. |
| 9      | When satisfied with your program:  
From the **D_100** graphic in Station, set the **CIRCULATE** and **SAFE** combo boxes to **OFF**.  
Change **SCM_D100** Mode Attribute to **Operator**, and change the state to **Idle**. |
| 10     | Inactivate **SCM_D100**. |
Use a Step to Start Another SCM

Objective

- Configure SCM_D100 to start the SCM_Pressure program

Prerequisites

- Experion PKS Server with all required CMs loaded
- D_100.htm graphic loaded on the Server (This is required at a later time to check the control strategy.)
- Control Builder running with one or two Project/Monitor tree windows open
- Excel Data Exchange open with the simulation spread sheet loaded
- All previous SCM labs in this section completed, tested, and verified

Introduction

In this section of the program, you will modify SCM_D100 to cause it to start SCM_Pressure (the first program you wrote).

The process described in this lab is totally fictitious and created only to demonstrate SCM features.

- Change the MODEATTR of SCM_Pressure to PROGRAM, and then start SCM_Pressure
- Check to see if 11_PC15 has reached 1240 Kpag
- Change the MODEATTR and MODE of selected points to be used later
- Check to see if 11_FC20 is closed
Continue with the Main Sequence

<table>
<thead>
<tr>
<th>✓ Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>For now, we are done with the Abort branch of the SCM. You will add these next steps and transitions under the <strong>SYNC_1</strong> block as shown below. Open Control Builder if not already opened.</td>
</tr>
</tbody>
</table>

![Abort Branch Diagram]

**ATTENTION**

Write expressions to start the program you built in a previous module.

Drag a transition from the library onto the project chart view of SCM_D100. Enter the following:

- **Transition Name:** CHECK_IDLE
- **Transition Description:** Your choice
- **Condition:** Create an expression that checks to see if **SCM_PRESSURE.STATE** is in the **IDLE** state.
- **Primary Gate:** As necessary
- **Secondary Gate:** As necessary
### Sequential Control Module - Lab

#### Use a Step to Start Another SCM

<table>
<thead>
<tr>
<th>✓ Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| **3** | Drag a STEP from the library into your project.  
Position it under the CHECK_IDLE transition.  
Add two Step Outputs.  
Enter the following:  
Step Name: **START_PRESS_P**  
Step Description: **START_PRESS_P**  
Output 1 Expression:  
Create an expression that will change SCM_PRESSURE.MODEATTR to PROGRAM.  
Output 1 Description: **SCM_Pressure Mode Attribute to PROGRAM**  
Output 2 Expression:  
Create an expression that will change SCM_PRESSURE.COMMAND to Start.  
Output 2 Description: **SCM_Pressure Start command** |
| **4** | Drag a transition from the library into your project.  
**NOTE:** One of the last actions that SCM_PRESSURE performs is changing the setpoint of PC15 to 1240 KPag.  
SCM_D100 should check that SCM_Pressure ran successfully by checking that PC15 PV is close to 1240.  
Transition Name: **CHECK_PC15**  
Create a transition condition verifies that SCM_Pressure has completed executing by checking that PC15’s PV is between 1230 and 1250 KPag.  
Add expressions as needed.  
All other values:  
Your choice |
<table>
<thead>
<tr>
<th>✓ Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 5     | Drag a STEP from the library into your project.  
Add 16 Step Outputs.  
Enter the following:  
Step Name:  
**CIRCULATE**  
Create the expressions to do the following:  
Change the **MODEATTR** of **FC28, FC20, LC16, HC41, HC44** to **PROGRAM**  
Change the **MODE** of **FC28, FC20, LC16, HC41, HC44** to **MAN**  
Change the **OP** of **FC28, FC20, LC16, HC41, HC44** to **0.0**  
Change **FLAGS.FLOW.PV** to **20.0**  
Enforce Order Option:  
**AllOutputs**  
All other values:  
Your choice  

**ATTENTION**  
These points will be used in a later module. |
| 6     | Drag a transition from the library into your project.  
Add expressions, as needed.  
Transition Name:  
**VALVE_CHECK**  
Create a Transition that assures **FC20** is closed.  
All other values:  
Your choice |
| 7     | Wire all transitions and steps. |
| 8     | Prepare to check your program:  
From the D_100 graphic in Station, set the **CIRCULATE** and **SAFE** to **OFF**.  
Turn all of the fans **ON**.  
In Control Builder, inactivate **SCM_D100**.  
Put **SCM_PRESSURE** Mode Attribute to Operator and the State in Idle. |
<table>
<thead>
<tr>
<th>✓ Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td><strong>Program Check:</strong>&lt;br&gt;Save, load, activate, and run SCM_D100.&lt;br&gt;Call the display <strong>D_100</strong> in <strong>Station</strong>.&lt;br&gt;Click on <strong>SCM_D100</strong> button to operate <strong>CIRCULATE</strong> and <strong>SAFE</strong> combo box.&lt;br&gt;If one or more Fan switches are in the OFF position, your program should follow the Abort sequence.&lt;br&gt;If all the fan switches are ON, then your program should start the SCM_PRESSURE program (Assuming it is in IDLE and ready to go). Next, the program should execute the CM changes indicated in Step 5, and then end after FC20 is closed.&lt;br&gt;Fix any problems you encounter.</td>
</tr>
<tr>
<td>10</td>
<td><strong>ATTENTION</strong>&lt;br&gt;It can take time for some points to change values due to tuning constants and the way simulation is implemented.</td>
</tr>
<tr>
<td></td>
<td><strong>When you are satisfied with your program:</strong>&lt;br&gt;Call up the D_100 graphic in Station and set the <strong>CIRCULATE</strong> and <strong>SAFE</strong> combo boxes to <strong>OFF</strong>.&lt;br&gt;In Control Builder, inactivate <strong>SCM_D100</strong>.&lt;br&gt;Put <strong>SCM_PRESSURE</strong> Mode Attribute to Operator and the State in Idle.</td>
</tr>
</tbody>
</table>
This page was intentionally left blank.
Use Recipe Values to Set Minimum Flow

Objective

- Configure this SCM to use recipe values to set a minimum flow

Prerequisites:

- Experion PKS Server with all required CMs loaded
- D_100.htm graphic loaded on the Server (This is required at a later time to check the control strategy.)
- Control Builder running with one or two Project/Monitor tree windows open
- Excel Data Exchange open with the simulation spread sheet loaded
- All previous SCM labs in this section completed, tested, and verified
## Turn Circulation on and Set Flow Level

<table>
<thead>
<tr>
<th>✓</th>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Open SCM_D100 in project view. Drag a STEP from the library into your project and place it under the VALVE_CHECK Transition. Add 15 Step Outputs Add a NEXTCOMP[2] pin to the bottom of this step. Enter the following: Step Name: CIRCULATE2 Create the expressions to do the following: Change the MODEATTR of AC12, FC28, FC19, PC15, PC16 to PROGRAM Change the MODE of AC12, FC28, PC15, PC16 to AUTO Change the MODE of FC19 to CAS Change the SP of AC12 to SCM_D100.RECTARGET[3] (This is the Recipe Target Value number 3) Change the SPs of PC15 and PC16 to a value that is equal to Recipe Target Value 2 plus 700.0 (Do this math within the Step Output expression. Do not change the recipe Target Value.) Turn HS62.FLAGA ON Turn HS68.FLAGA OFF Enforce Order Option: AllOutputs All other values: Your choice</td>
<td></td>
</tr>
<tr>
<td>✓ Step</td>
<td>Action</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Wire the output of CIRCULATE2 NEXTCOMP[2] pin back to the input of the VALVE_CHECK Transition. You will also need to temporarily wire the NEXTCOMP[1] pin back to the input of the VALVE_CHECK Transition so that you can do the program check in the next step.</td>
<td></td>
</tr>
</tbody>
</table>

![Diagram showing wire connections between VALVE_CHECK and CIRCULATE2]

| 3      | Program Check List:  
Save, load, activate and run your program.  
If one or more fan switches are in the OFF position, your program should follow the Abort sequence.  
If all the fan switches are ON, then your program should start the SCM_PRESSURE program (assuming it is in IDLE and ready to go). Next, the program should execute the CM changes indicated in the Circulate step.  
Once your program completes the CIRCULATE2 Step, it will branch back to the VALVE_CHECK Transition continuously. This allows you to set the circulate flow amounts by adjusting the Recipe Target Values. |

| 4      | When satisfied with your program:  
Call up the D_100 graphic in Station and set the CIRCULATE and SAFE combo boxes to OFF.  
In Control Builder, inactivate SCM_D100.  
Put SCM_PRESSURE Mode Attribute to Operator and the State in Idle. |
This page was intentionally left blank.
Interface an SCM with a Graphic Textbox

Objective

- Configure the SCM to accept input values from the Process Flow textbox on the D_100 graphic

Prerequisites

- Experion PKS Server with all required CMs loaded
- D_100.htm graphic loaded on the Server (This is required at a later time to check the control strategy.)
- Control Builder running with one or two Project/Monitor tree windows open
- Excel Data Exchange open with the simulation spread sheet loaded
- All previous SCM labs in this section completed, tested, and verified

Introduction

In this section of the SCM, you will write programming to tie the value in the Process Flow textbox (on the D_100 graphic) to the SPs of several controllers.

The process described in this lab is totally fictitious and created only to demonstrate SCM features.

- Create algorithms that convert the Process Flow textbox to a SP value in engineering unit, for several controllers.
## Add the Process Sequence

<table>
<thead>
<tr>
<th>✓  Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1      | Open the SCM_D100 project chart view. Delete the wire connection from CIRCULATE2 NEXTCOMP[1] to the input of the VALVE_CHECK transition. Leave the NEXTCOMP[2] connection as is. Drag a transition from the library into your project and place it under the CIRCULATE2 step. Add expressions as needed. Transition Name: **PROCESS** Create a transition that checks to see if **FLAGS.PROCESS** is ON. All other values:  
Your choice |
| 2      | On the D-100 display, is a field named “Process Flow” (the actual parameter is Flags.Flow.PV). The operator can enter a “Process Flow” from 0 – 100. This value represents the percent of range in which the process will operate. For example, if the operator enters 75, the SP of a tag will be changed to 75% of range. This step, and the next step will accomplish that.  
  a. Drag a STEP from the library into your project and place it under the PROCESS Transition.  
  b. Add Step Outputs as necessary  
    (Count the number of expressions required by this step (2) AND the next step (4).)  
  c. Enter the following:  
Step Name: **RUN_PROCESS**  
Create the expressions to do the following:  
  - Change the **MODEATTR** of FC20, LC16 to PROGRAM  
  - Change the **MODE** of FC20 to CAS  
  - Change the **MODE** of LC16 to AUTO  
  - Turn **HS68 FlagA ON**. (Use 11_HS68.FlagA.PVFL) |
<table>
<thead>
<tr>
<th>✓</th>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>Create expressions which will calculate values based on &quot;Process Flow&quot; (Flags.Flow.PV) as shown below:</td>
</tr>
</tbody>
</table>

PC15 and PC16 are reverse acting controllers. Use the following expressions to calculate their SPs:

\[
11_{-}PC15.PIDA.SP := (1050.0 - (11_{-}PC15.PIDA.PVEUHI - 11_{-}PC15.PIDA.PVEULO) \times (FLAGS.FLOW.PV/100.0)) + 11_{-}PC15.PIDA.PVEULO
\]

\[
11_{-}PC16.PIDA.SP := (1050.0 - (11_{-}PC16.PIDA.PVEUHI - 11_{-}PC16.PIDA.PVEULO) \times (FLAGS.FLOW.PV/100.0)) + 11_{-}PC16.PIDA.PVEULO
\]

FC28 and LC16 are direct acting controllers. Use the following expressions to calculate their SPs:

\[
11_{-}FC28.PIDA.SP := ((11_{-}FC28.PIDA.PVEUHI - 11_{-}FC28.PIDA.PVEULO) \times (FLAGS.FLOW.PV/100.0)) + 11_{-}FC28.PIDA.PVEULO
\]

\[
11_{-}LC16.PIDA.SP := ((11_{-}LC16.PIDA.PVEUHI - 11_{-}LC16.PIDA.PVEULO) \times (FLAGS.FLOW.PV/100.0)) + 11_{-}LC16.PIDA.PVEULO
\]

Calculate AC12 SP based on the SP High and SP Low limits (rather than the range of the tag). Use the following expression to calculate the SP:

\[
11_{-}AC12.PIDA.SP := ((11_{-}AC12.PIDA.SPHILM - 11_{-}AC12.PIDA.SPLOLM) \times (FLAGS.FLOW.PV/100.0)) + 11_{-}AC12.PIDA.SPLOLM
\]

Use the following expressions to vary the OP of 11_HC41 and 11_HC44 in proportion to FLAGS.FLOW:

\[
11_{-}HC41.AUTOMANA.OP := ((11_{-}HC41.AUTOMANA.XEUHI - 11_{-}HC41.AUTOMANA.XEULO) \times (FLAGS.FLOW.PV/100.0)) + 11_{-}HC41.AUTOMANA.XEULO
\]

\[
11_{-}HC44.AUTOMANA.OP := ((11_{-}HC44.AUTOMANA.XEUHI - 11_{-}HC44.AUTOMANA.XEULO) \times (FLAGS.FLOW.PV/100.0)) + 11_{-}HC44.AUTOMANA.XEULO
\]
<table>
<thead>
<tr>
<th>✓ Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Wire the <strong>RUN_PROCESS NEXTCOMP[1]</strong> pin back to the input of the <strong>PROCESS</strong> Transition.</td>
</tr>
</tbody>
</table>

![Diagram](image)
<table>
<thead>
<tr>
<th>✓ Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td><strong>Program Check:</strong>&lt;br&gt;Save, load, activate, and run your program.&lt;br&gt;  If one or more fan switches are in the <strong>OFF</strong> position, your program should follow the Abort sequence.&lt;br&gt;  If all the fan switches are <strong>ON</strong>, then your program should start the <strong>SCM_PRESSURE</strong> program (assuming it is in IDLE and ready to go).&lt;br&gt;  Once your program completes the CIRCULATE2 step, it will branch back to the VALVE_CHECK transition continuously until the PROCESS flag is turned ON.&lt;br&gt;  When your program reaches the CIRCULATE2 step, select and change <strong>PROCESS</strong> combo box to <strong>ON</strong>.&lt;br&gt;  You should now be able to select and change the amount of flow in the <strong>PROCESS FLOW</strong> textbox in the lower right of the D_100 graphic, and see the process respond accordingly. Click on <strong>FLOW</strong> button to open popup window.&lt;br&gt;  Set Process Flow to 30%. Verify that the SPs and OPs of the tags were calculated and stored successfully:&lt;br&gt;  PC15.SP and PC16.SP = 1435&lt;br&gt;  FC28.SP = 46.50&lt;br&gt;  LC16.SP = 30&lt;br&gt;  AC12.SP = 0.6&lt;br&gt;  HC41.OP and HC44.OP = 30  (look on Debutanizer_123)&lt;br&gt;  Set Process Flow to 50%. Verify that the SPs and OPs of the tags were calculated and stored successfully:&lt;br&gt;  PC15.SP and PC16.SP = 1225&lt;br&gt;  FC28.SP = 77.50&lt;br&gt;  LC16.SP = 50&lt;br&gt;  AC12.SP = 1.0&lt;br&gt;  HC41.OP and HC44.OP = 50  (look on Debutanizer_123)&lt;br&gt;  Try setting the Process Flow to other values. Setting the flow above 75% will put a flame in the Flare Cam screen. Setting the flow above 95% will cause the compressor to explode.</td>
</tr>
<tr>
<td>6</td>
<td>When you are satisfied with your program:&lt;br&gt;  Call up the D_100 graphic in Station. Click on <strong>SCM_D100</strong> button and set the <strong>CIRCULATE, SAFE, and PROCESS</strong> combo boxes to <strong>OFF</strong>.&lt;br&gt;  In Control Builder, inactivate <strong>SCM_D100</strong>.&lt;br&gt;  Put <strong>SCM_PRESSURE</strong> Mode Attribute to Operator and the State in Idle.</td>
</tr>
</tbody>
</table>
This page was intentionally left blank.
**R410: Use DATA Block (Formula Parameter) to set minimum Flow**

**Objective**

- Modify SCM_D100 to use Formula Parameter from DATA Block to set a minimum flow

**Prerequisites:**

- Experion PKS Server with all required CMs loaded
- D_100.htm graphic loaded on the Server (This is required at a later time to check the control strategy.)
- Control Builder running with one or two Project/Monitor tree windows open
- Excel Data Exchange open with the simulation spread sheet loaded
- All previous SCM labs in this section completed, tested, and verified
- SCM_D100 is configured and tested
Turn Circulation on and Set Flow Level using Formula Parameter

<table>
<thead>
<tr>
<th>✓</th>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>Open Control Builder if already not opened and to create a new Phase Block, <strong>File → New → Type → Phase Block</strong></td>
</tr>
</tbody>
</table>

![Image of Control Builder](image.png)
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 2    | Give **Library Name** and **Type Name** as follows.  
Library Name – CUSTOM_DB  
Type Name – TARGET_FLOW  
Click **OK**. This creates Phase block with name TARGET_FLOW. It appears in Library. |

![Library & Type](image1.png)

![Library - Containment](image2.png)
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>In the Parameter Definition Editor, create custom Formula Parameters as follows,&lt;br&gt;Parameter name: <strong>TARGET_FLOW_1</strong>&lt;br&gt;Parameter Description: <strong>Target Flow value 1</strong>&lt;br&gt;Data type: <strong>INT32</strong>&lt;br&gt;Default Value: <strong>47</strong>&lt;br&gt;Min. Value: <strong>0</strong>&lt;br&gt;Max. Value: <strong>100</strong>&lt;br&gt;&lt;br&gt;Parameter name: <strong>TARGET_FLOW_2</strong>&lt;br&gt;Parameter Description: <strong>Target Flow value 2</strong>&lt;br&gt;Data type: <strong>INT32</strong>&lt;br&gt;Default Value: <strong>850</strong>&lt;br&gt;Min. Value: <strong>0</strong>&lt;br&gt;Max. Value: <strong>1750</strong>&lt;br&gt;&lt;br&gt;Parameter name: <strong>TARGET_FLOW_3</strong>&lt;br&gt;Parameter Description: <strong>Target Flow value 3</strong>&lt;br&gt;Data type: <strong>INT32</strong>&lt;br&gt;Default Value: <strong>1</strong>&lt;br&gt;Min. Value: <strong>0</strong>&lt;br&gt;Max. Value: <strong>6</strong>&lt;br&gt;For all formula parameters <strong>Enabled</strong> is checked and Access Lock is <strong>OPERATOR</strong>.</td>
</tr>
<tr>
<td>4</td>
<td><strong>Save</strong> and <strong>Close</strong> Phase Block.</td>
</tr>
<tr>
<td>5</td>
<td>Right click on <strong>SCM_D100</strong> and select <strong>Module Properties</strong>.</td>
</tr>
</tbody>
</table>

**ATTENTION**

Parameter Description is essential as only parameter description is available while viewing Data Block properties.
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Select <strong>Parameters</strong> tab and browse the required Phase Block in <strong>Data Block</strong> field.</td>
</tr>
<tr>
<td></td>
<td><img src="image" alt="Screenshot of selecting Phase Block in Data Block field" /></td>
</tr>
<tr>
<td></td>
<td><strong>ATTENTION</strong></td>
</tr>
<tr>
<td></td>
<td>Phase block used with SCM is known as Data Block.</td>
</tr>
<tr>
<td></td>
<td>All available DATA Blocks are listed for selection.</td>
</tr>
<tr>
<td>7</td>
<td>Click <strong>OK</strong>.</td>
</tr>
<tr>
<td></td>
<td>Data Block will get configured for SCM as shown in screen capture.</td>
</tr>
<tr>
<td></td>
<td><img src="image" alt="Screenshot of configured Data Block" /></td>
</tr>
<tr>
<td>8</td>
<td>Click <strong>OK</strong> and Close <strong>SCM_D100</strong> property window.</td>
</tr>
</tbody>
</table>
### Sequential Control Module - Lab

**R410: Use DATA Block (Formula Parameter) to set minimum Flow**

<table>
<thead>
<tr>
<th>✓</th>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 9 | Open SCM_D100 in project view. Go to step name **CIRCULATE2**. Change the following output expression as shown below (Output expression 11,12,13), | ![Table](Old Output Expressions | New Output Expressions)

<table>
<thead>
<tr>
<th>Old Output Expressions</th>
<th>New Output Expressions</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 11_PC15.PIDA.SP:=SCM_D100.RECTARGET[2]+700 )</td>
<td>( 11_PC15.PIDA.SP:=SCM_D100.DA TA.TARGET_FLOW_2.VALUE+700 )</td>
</tr>
</tbody>
</table>

Keep other expressions as it is.

**Enforce Order Option:**

- **AllOutputs**
- All other values: 
  - Your choice

**TIP**

Using Formula Parameters we get same result as that of Recipe values.

<table>
<thead>
<tr>
<th>✓</th>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 10 | Program Check List: Save, load, activate and run your program. If one or more fan switches are in the **OFF** position, your program should follow the Abort sequence. If all the fan switches are **ON**, then your program should start the **SCM\_PRESSURE** program (assuming it is in IDLE and ready to go). Next, the program should execute the CM changes indicated in the Circulate step. Once your program completes the CIRCULATE2 Step, it will branch back to the **VALVE\_CHECK** Transition continuously. This allows you to set the circulate flow amounts by adjusting the Formula Parameters. | **TIP**

Do not change Process Combox
Step | Action
--- | ---
11 | Call SCM_D100 detail display on Station.
12 | Select Formula Tab. Enter the 700 in the Default value of Formula Parameter TARGET_FLOW_2 of Data Block.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Parameter Description</th>
<th>Default Value</th>
<th>Current Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TARGET_FLOW_1</td>
<td>Target Flow Value 1</td>
<td>47</td>
<td>47</td>
</tr>
<tr>
<td>TARGET_FLOW_2</td>
<td>Target Flow Value 2</td>
<td>350</td>
<td>350</td>
</tr>
<tr>
<td>TARGET_FLOW_3</td>
<td>Target Flow Value 3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

13 | Open detail display of 11_PC15 and 11_PC16. You can see that SP of 11_PC15 and 11_PC16 are changing as per Formula Parameter of Data Block.

TIP
As per Expression
11_PC15.PIDA.SP:=SCM_D100.DATA.TARGET_FLOW_2.VALUE+700, SP for 11_PC15 and 11_PC16 are set to 1400. (700 + 700)

14 | In Control Builder, open SCM_D100 in Chart view and go to Edit → DATA Block → DATA Block Properties

15 | Note the Value for each Formula Parameter. Using this option you can view the current value of Formula Parameter from Control Builder for respective SCM.
When satisfied with your program:
Call up the D_100 graphic in Station and set the **CIRCULATE** and **SAFE**
combo boxes to **OFF**.
In Control Builder, inactivate **SCM_D100**.
Put **SCM_PRESSURE** Mode Attribute to Operator and the State in **Idle**.

Now we will create Activity for **SCM_D100**. We will see how to change SP of **11_PC15** and **11_PC16** through Activity using Formula Parameter.

Inactivate **SCM_D100**.

Open **Module Properties** of **SCM_D100** from project window and select Main tab.

On the Main tab, select the CEE Restart Option "**ALWAYS COLD**".

**ATTENTION**

When SCMs are associated with Activities warm restart option is not supported.
<table>
<thead>
<tr>
<th>✓ Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>Select Activities tab. In Activities Tab, Select <strong>Type of the Activity</strong> as <strong>Procedure</strong>. Select <strong>Activity Creation option</strong> as <strong>User or App</strong>.</td>
</tr>
</tbody>
</table>

**TIP**

**SCM_Pressure** is associated with Activity Type **Procedure**. Activity Type for both SCM must be same as **SCM_Pressure** is starting from **SCM_D100**.
### Sequential Control Module - Lab

**R410: Use DATA Block (Formula Parameter) to set minimum Flow**

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>22</strong></td>
<td>Click on <strong>Arbitration</strong> tab. Give <strong>Maximum Waiting Requesters</strong> as 10 (maximum number cab be assignable is 10). Click <strong>OK</strong>.</td>
</tr>
<tr>
<td><strong>23</strong></td>
<td>Save, Load and Activate <strong>SCM_D100</strong>.</td>
</tr>
<tr>
<td><strong>24</strong></td>
<td>Go to Activity summary display on Station. <strong>View → Activities</strong></td>
</tr>
</tbody>
</table>

**TIP**

Using this option, you can view recent 10 pending requests for respective SCM, CM on Station or from Control Builder. Control Strategy must be active to view pending requests.
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>Click on <strong>New</strong> at the right corner and select <strong>New Procedure</strong>.</td>
</tr>
</tbody>
</table>

**TIP**
User must select correct type of Activity while creating a new Activity.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>Select <strong>SCM_D100</strong> and click on <strong>Create</strong>.</td>
</tr>
</tbody>
</table>

**TIP**
Activity can be created manually only if **Activity Creation option** is selected as **User or App** or **All**.

**SCM_Pressure** is not available in the list as **Activity Creation option** is selected as **Auto** for **SCM_Pressure**.
## Step 27

A new Activity gets created in Activity summary.

<table>
<thead>
<tr>
<th>Batch ID</th>
<th>Activity Entity</th>
<th>Public Name</th>
<th>Asset</th>
<th>Type</th>
<th>Description</th>
<th>Stage</th>
<th>State</th>
<th>Status</th>
<th>Status Des</th>
</tr>
</thead>
<tbody>
<tr>
<td>59100E93</td>
<td>SCM_D100</td>
<td>SCM_D100</td>
<td>C11</td>
<td>Procedure</td>
<td>Circulate / Process</td>
<td>PreExecution</td>
<td>Ok</td>
<td>Ok</td>
<td></td>
</tr>
</tbody>
</table>

**ATTENTION**

You can create an Activity manually, if the Activity Creation option is selected as User or app or All. Logged in Operator must have access for Parent asset of SCM and then only operator is able to create an Activity.

## Step 28

Select the Activity from summary.

<table>
<thead>
<tr>
<th>Batch ID</th>
<th>Activity Entity</th>
<th>Public Name</th>
<th>Type</th>
<th>Description</th>
<th>Stage</th>
<th>State</th>
<th>Status</th>
<th>Status Des</th>
</tr>
</thead>
<tbody>
<tr>
<td>59100E93</td>
<td>SCM_D100</td>
<td>SCM_D100</td>
<td>Procedure</td>
<td>Circulate / Process</td>
<td>PreExecution</td>
<td>Ok</td>
<td>Ok</td>
<td></td>
</tr>
</tbody>
</table>

**TIP**

Activity remains in PreExecution stage until Start command is not issued for respective activity.
Click on Formula Parameters.
Enter value 200 in New Value for Target Flow Value 2 and click **Apply**.

**TIP**

Value 200 in now available in **Control Recipe Value** after clicking on Apply. (200 + 700 = 900)
This the new set point for 11_PC15 and 11_PC16 as per Output Expression.
**Step 30**

Give a **Start** command to Activity.

<table>
<thead>
<tr>
<th>Batch ID</th>
<th>Activity Entity</th>
<th>Public Name</th>
<th>Asset</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>50103E91E24</td>
<td>SCM_D100</td>
<td>SCM_D100</td>
<td>C11</td>
<td>Procedure</td>
<td>Circulate / P</td>
</tr>
</tbody>
</table>

![Start Hold Stop Remove Abort]

**Tip**

The **Start** command can be issued from Activity summary display, Faceplate, Chart view on Station and directly from Control Builder monitoring window.

---

**Step 31**

Call up the D_100 graphic in Station. Click on SCM_D100 button and set the **SAFE** combo box to **ON**.

**Step 32**

Call up detail display of **11_PC15** and **11_PC16** on Station. Verify SP is set to 900.

**Step 33**

Go to Activity summary display, you can see activities for **SCM_Pressure** (created in previous lab) and **SCM_D100**.
When satisfied with your program:

Call up the D_100 graphic in Station. Click on SCM_D100 button and set SAFE combo box to OFF.

From activity summary display, Abort and Reset SCM_D100. Put SCM_D100 to Idle state.

Put SCM_PRESSURE Mode Attribute to Operator and the State in Idle.

As SCM goes into Idle state, respective Activity disappears from summary.

Now we will see how we can create multiple Activities for same SCM and give them Start command simultaneously.

Create three new activities for SCM_D100. Give Formula Parameters for each Activity as per choice.

Give Start command to all activities.

ATTENTION

Only one Activity will start executing (which is first commanded to Start). All other activities show status Acquiring Resource.
37 Call up the D_100 graphic on Station and set **SAFE** combo box to **ON**. You can notice that new SP is assigned to **11_PC15** and **11_PC16**.

<table>
<thead>
<tr>
<th>Activity Entry</th>
<th>Public Name</th>
<th>Asset</th>
<th>Type</th>
<th>Description</th>
<th>Stage</th>
<th>State</th>
<th>Status</th>
<th>Status Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCM_Pressure</td>
<td>SCM_Pressure</td>
<td>C11</td>
<td>Proc.</td>
<td>Pressure on T-100</td>
<td>Executing</td>
<td>Ok</td>
<td>Ok</td>
<td></td>
</tr>
<tr>
<td>SCM_D100</td>
<td>SCM_D100</td>
<td>C11</td>
<td>Batch</td>
<td>Circulate / Process</td>
<td>Running</td>
<td>Ok</td>
<td>Ok</td>
<td></td>
</tr>
<tr>
<td>SCM_D100</td>
<td>SCM_D100</td>
<td>C11</td>
<td>Batch</td>
<td>Circulate / Process</td>
<td>PreExecution</td>
<td>Info</td>
<td>Acquiring Resource</td>
<td></td>
</tr>
<tr>
<td>SCM_D100</td>
<td>SCM_D100</td>
<td>C11</td>
<td>Batch</td>
<td>Circulate / Process</td>
<td>PreExecution</td>
<td>Info</td>
<td>Acquiring Resource</td>
<td></td>
</tr>
</tbody>
</table>
## Step 38

Call detail display of CEE on Station. Select **Statistics** tab.

Note down the values for following:

- Num Activity Currently Running: 
- Max Number Activities: 
- Total Activity Capacity: 

### Config Details

<table>
<thead>
<tr>
<th>Peer Connections</th>
<th>Peer Communications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server History</td>
<td>Peer Configuration</td>
</tr>
</tbody>
</table>

- **Max Total Responder Rate**: 5119
- **Peer Responder Rate**: 0
- **Max Peer Responder Rate**: 0
- **Display Responder Rate**: 649.0532
- **Max Display Responder Rate**: 5119

### Activity Statistics

- **Num Activities Currently Running**: 4
- **Max Number Activities**: 4
- **Total Activity Capacity**: 100

### TIP

This information you can view in Control Builder also.
39 Call detail display of SCM_D100 on Station. Click on Arbitration Tab. You will get number of pending requests for acquiring the resource.

<table>
<thead>
<tr>
<th>Command</th>
<th>Mode</th>
<th>Mode Attr</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Inactive</td>
<td>Start</td>
</tr>
</tbody>
</table>

40 Abort and Reset SCM_D100 when an Activity for SCM_Pressure shows state Complete. Put SCM_Pressure and SCM_D100 to Idle.

TIP

The Reset command can be issued from Activity summary display, Faceplate, Chart view on Station and directly from Control Builder monitoring window.
Sequential Control Module - Lab
R410: Use DATA Block (Formula Parameter) to set minimum Flow

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>41</td>
<td>As <strong>Start</strong> command is already issued to second Activity and <strong>SAFE</strong> combo box is still set to <strong>ON</strong> from D_100 graphic, SCM_D100 will start executing immediately.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Batch ID</th>
<th>Activity Entity</th>
<th>Public Name</th>
<th>Asset</th>
<th>Type</th>
<th>Description</th>
<th>Stage</th>
<th>State</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>90168X118</td>
<td>SCM_D100</td>
<td>SCM_D100</td>
<td>C11</td>
<td>Procedure</td>
<td>Pressure on T-100</td>
<td>Executing</td>
<td>Complete</td>
<td>Ok</td>
</tr>
<tr>
<td>90168X118</td>
<td>SCM_D100</td>
<td>SCM_D100</td>
<td>C11</td>
<td>Procedure</td>
<td>Circulate / Process</td>
<td>Executing</td>
<td>Running</td>
<td>Ok</td>
</tr>
</tbody>
</table>

New SP is assigned to the **11_PC15** and **11_PC16** as per values in Formula Parameters of current activity.

Same way third activity will also start executing after changing **SCM_Pressure** and **SCM_D100** to **Idle** state.

| 42   | When satisfied with your program: |
|      | Call up the D_100 graphic in Station. Click o SCM_D100 button and set the **CIRCULATE** and **SAFE** combo boxes to **OFF**. |
|      | In Control Builder, inactivate **SCM_D100**. |
|      | Put **SCM_PRESSURE** Mode Attribute to Operator and the State in Idle. |
This page was intentionally left blank.