

Connected Components Workbench Task-based Quick Start

QS#AF04 - Creating a New Structured Text Program

LISTEN.
THINK.
SOLVE.®

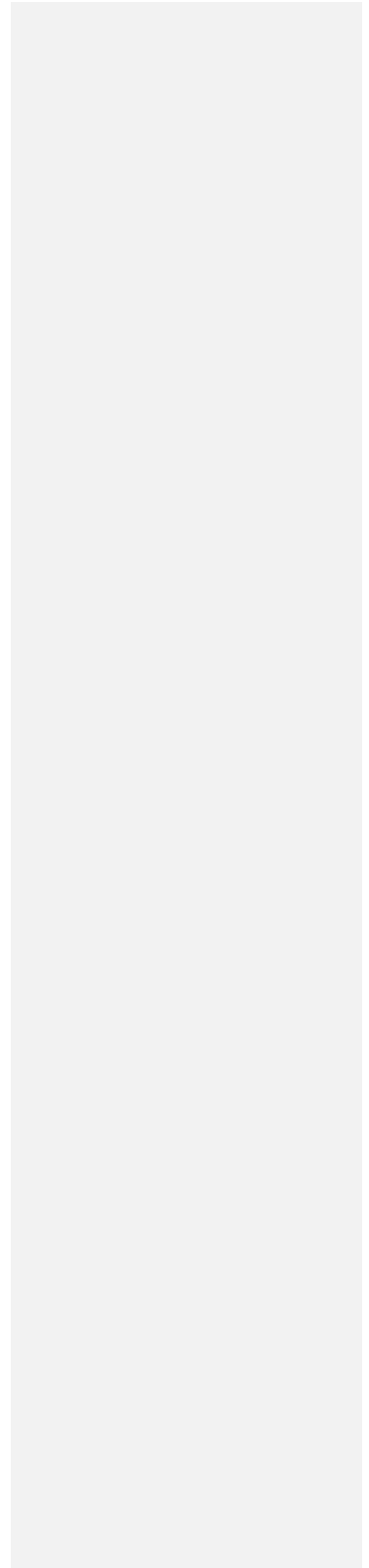
 Allen-Bradley • Rockwell Software

**Rockwell
Automation**

Copyright © 2009 Rockwell Automation, Inc.

Hardware & Software Versions Used to Develop This Quick Start

- CCW Release 1, Build 59
- 2080-LC30-16QWB v1.112



Creating a New Structured Text Program

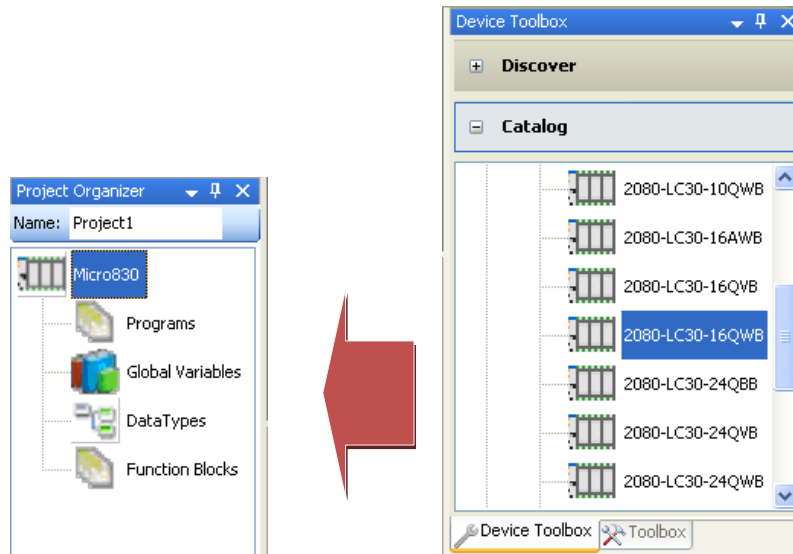
This quick start will show you how to create a new structured text program for creating menu selections and simple mathematical calculations.

1. Start the Connected Component Workbench for the Start Menu: **Start → All Programs → Rockwell Automation → CCW → Connected Components Workbench.**



Alternatively, double click on the shortcut on the Desktop .

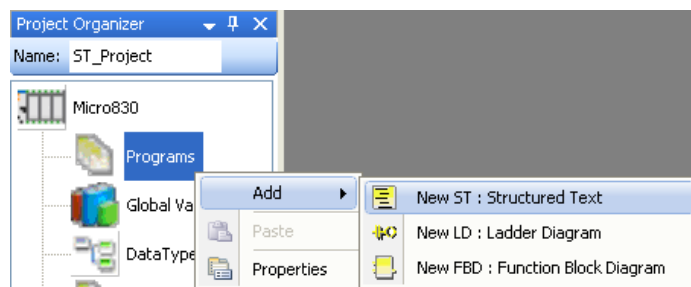
- At the Connected Component Workbench window, drag **2080-LC30-16QWB** from the **Device Toolbox Catalog** window into the **Project Organizer** window. A new project will be created.



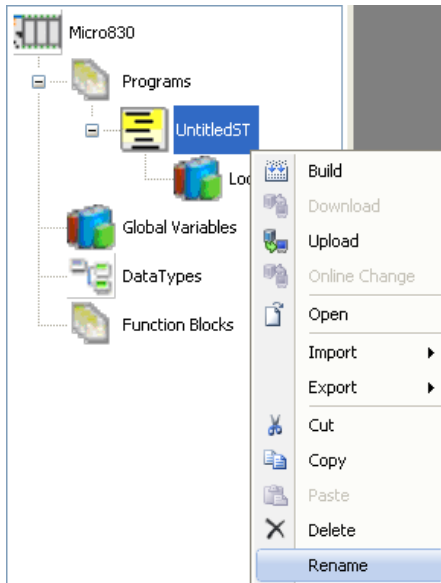
- At the **Name** field, under the **Project Organizer**, enter **ST_Program**.



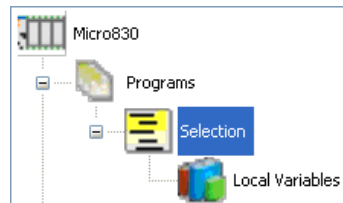
- Under the **Project Organizer**, right click on the **Programs** select **Add** and select **New ST: Structured Text**.



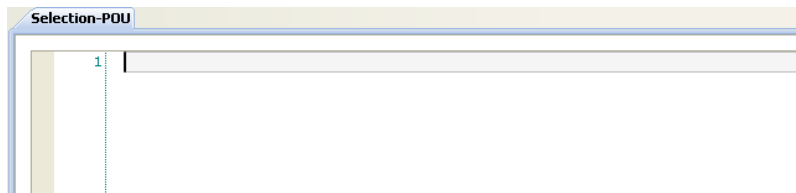
- Right click on **UntitledST** and select **Rename**:



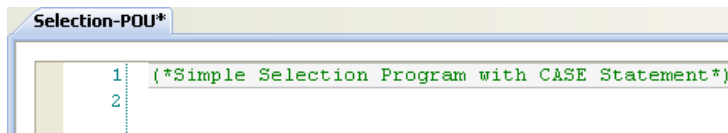
- Type **Selection** and Enter:



- Double click on **Selection** within the **Project Organizer** to start editing the Structured Text program.
- Click at the Line no. "1" at the **Selection-POU*** tab.



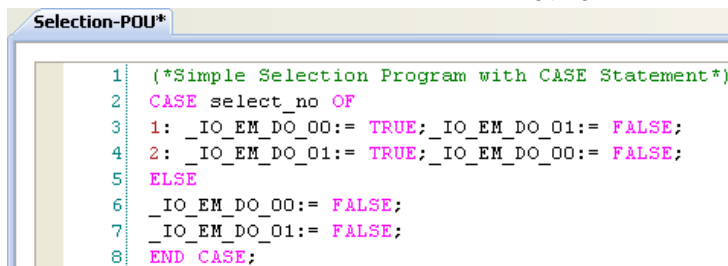
9. Enter the following sentence “(*Simple Selection Program with CASE Statement*)”, then hit enter.



```
1 (*Simple Selection Program with CASE Statement*)
2
```

Note: For entering comments use “(* comments *)”

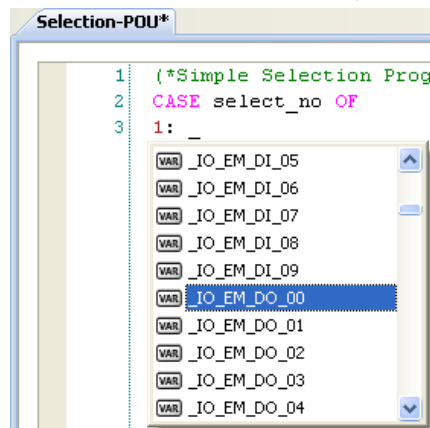
10. Click at Line no. "2" at the **Selection-POU*** tab, enter the following program.



```
1 (*Simple Selection Program with CASE Statement*)
2 CASE select_no OF
3 1: _IO_EM_DO_00:= TRUE; IO_EM_DO_01:= FALSE;
4 2: _IO_EM_DO_01:= TRUE; _IO_EM_DO_00:= FALSE;
5 ELSE
6 _IO_EM_DO_00:= FALSE;
7 _IO_EM_DO_01:= FALSE;
8 END_CASE;
```

Note: All Structured Text Reserve word will be represented in magenta, and comments will be represented in Green.

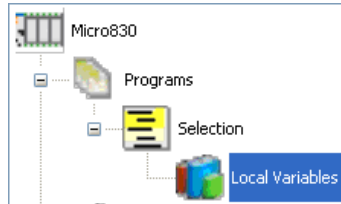
When entering the IO variable, we are able to select from the pull down menu as shown



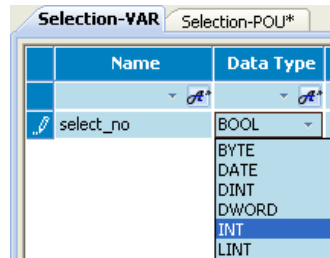
```
1 (*Simple Selection Prog
2 CASE select_no OF
3 1: _
```

For Boolean expression, True is “1” and False is “0”.

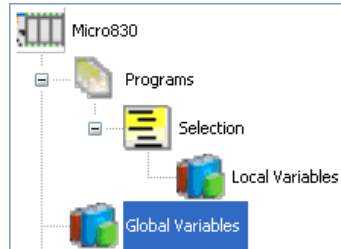
11. Double click on the **Local Variables** under the Selection programs to define a new variable.



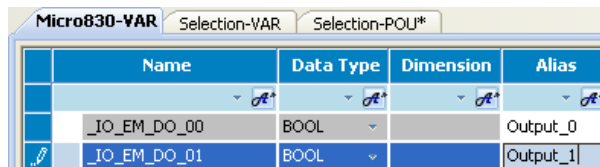
12. Create an **integer** variable **select_no** as shown:



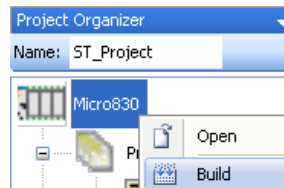
13. At the Project Organizer, double click on the **Global Variables** to create the Alias for the outputs.



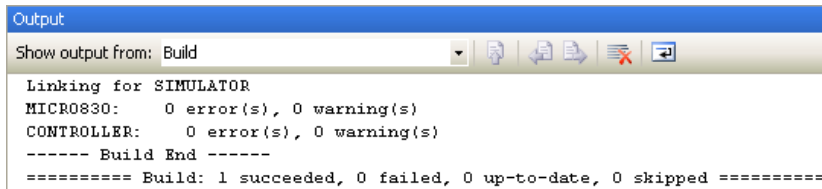
14. At **Micro830-VAR** tab, enter **Output_0** at **Alias** for **_IO_EM_DO_00** and **Output_1** at **Alias** for **_IO_EM_DO_01**.



15. Finally, build and save the structured text programming. Right click on the Micro830 icon in **Project Organizer** and select **Build**.



16. At the **Output** window at the bottom center of the screen, the build should show succeeded.

The image shows an 'Output' window. The title bar says 'Output'. Below the title bar, there is a dropdown menu set to 'Build' and several icons. The main area contains the following text:

```
Linking for SIMULATOR
MICRO830:    0 error(s), 0 warning(s)
CONTROLLER:  0 error(s), 0 warning(s)
----- Build End -----
===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
```

Click on **Save** icon  to save your work.

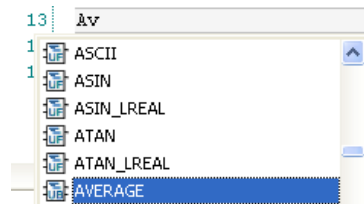
Inserting Function block a Structured Text Program

This section will show you how to insert a function block in the existing Structured Text Program.

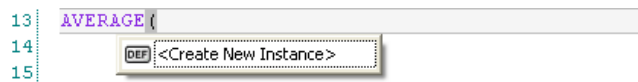
1. Double click on the **Selection**, to edit.
2. At Line 10 of the **Selection-POU*** tab, enter the following sentences

```
9 |
10 | IF _IO_EM_DO_00 THEN
11 |   i := a*b*c;
12 |   ELSE IF _IO_EM_DO_01 THEN
```

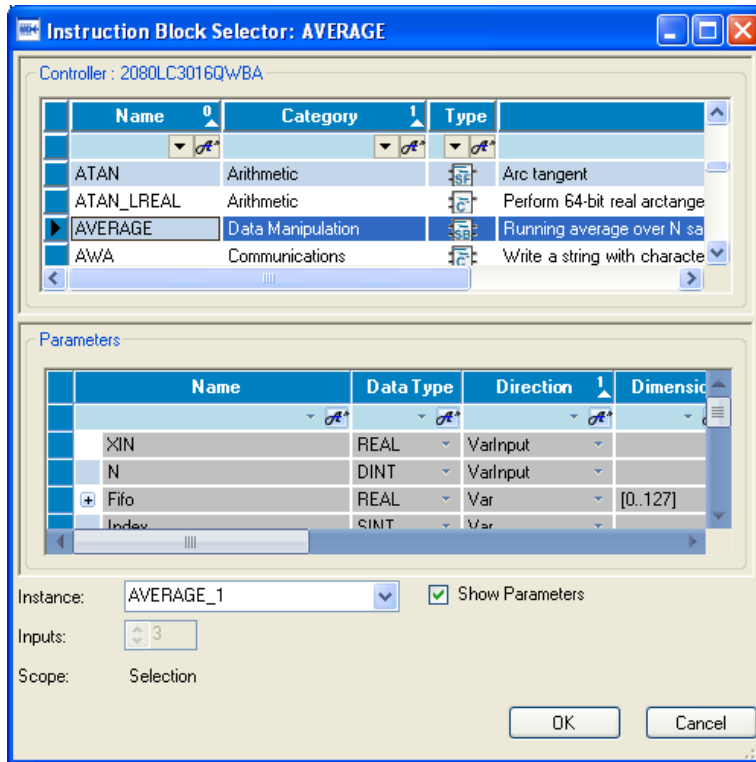
3. At line 13 of the **Selection-POU*** tab, enter “AV” and select **AVERAGE** from the pull down menu.



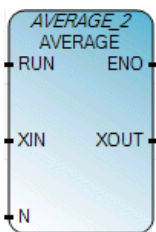
4. Then key in “ (“ the following pull down menu will appear. Select the **<Create New Instance>**



- The following dialog box will appear, AVERAGE_1 is created.



Note: 3 Inputs are required for Average function block, similar to Ladder Logic Representation.



RUN, XIN,N parameter will be required.

6. Click **OK** to create an instance. When entering the instance, the popup box will indicate the parameter needed for the Function block.

```
10 IF _IO_EM_DO_00 THEN
11 i:= a*b*c;
12 ELSE IF _IO_EM_DO_01 THEN
13 AVERAGE_1(
```

void **AVERAGE_1**(BOOL RUN, REAL XIN, DINT N)
Type : AVERAGE, Running average over N samples

7. Please end the parameter as shown:

'AVERAGE_1(_IO_EM_DO_01,a,3)'

Where:

RUN = _IO_EM_DO_01

XIN = a

N = 3

Kommentar [FDR1]: Just changed the formatting a little to make it more clear what text to enter.

8. Then assign the output of the AVERAGE to j, as per shown. Close the IF statement with END_IF.

```
9
10 IF _IO_EM_DO_00 THEN
11 i:= a*b*c;
12 ELSE IF _IO_EM_DO_01 THEN
13 AVERAGE_1(_IO_EM_DO_01, a, 3);
14 j:= AVERAGE_1.XOUT;
15 END_IF;
16 END_IF;
```

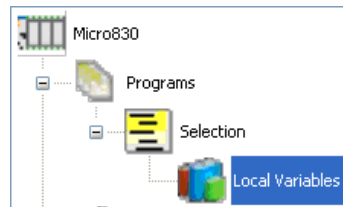
Notes:

- The mathematical equation can be expressed by entering it as is. If doing the calculation in ladder, you might need a few function blocks to complete the equation.

Example: $i := a + b + c$; or $\text{circumference} := 2 * 3,142 * r$; (with r is the variable) or $r := \text{circumference} / (2 * 3,142)$;

- When using IF statement, we must also close with an END_IF, in the case if there is an ELSE_IF statement used, we must also close the ELSE_IF statement with END_IF.

9. In completion of writing the program, variables used must be created. Double click on the **Local Variables** under the Selection programs to create variable.



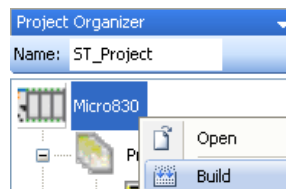
10. Create the following variables for the program

Name	Data Type	Initial Value
a	Real	0.0
b	Real	1.5
c	Real	3.142
i	Real	2.0
j	Real	0.0

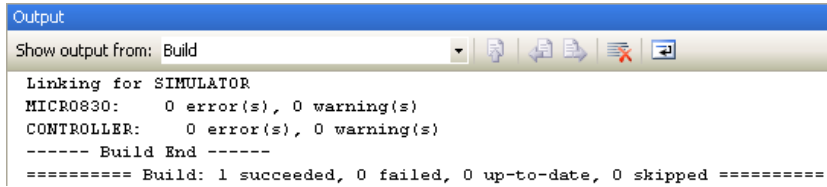
The Selection-VAR tab should look like the following:

Name	Data Type	Dimension	Alias	Comment	Initial Value
select_no	INT				
i	REAL				0.0
a	REAL				1.5
b	REAL				3.142
c	REAL				2.0
j	REAL				
AVERAGE_1	AVERAGE				...

11. Finally, build and save the structure text programming. Right click on the Micro830 icon in **Project Organizer** and select **Build**.



12. At the **Output** window at the bottom center of the screen, the build should show succeeded.



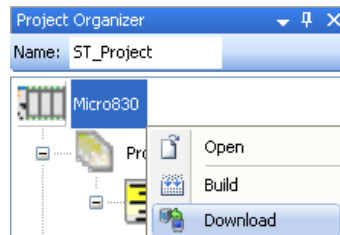
```
Output
Show output from: Build
Linking for SIMULATOR
MICRO830: 0 error(s), 0 warning(s)
CONTROLLER: 0 error(s), 0 warning(s)
----- Build End -----
==== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
```

Click on **Save** icon  to save your work.

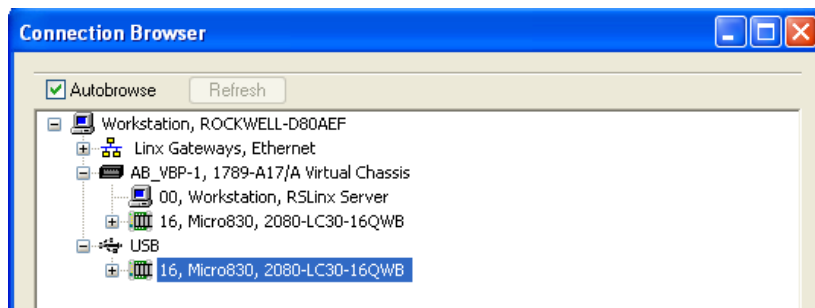
Testing the Function Block Program

This section will show you how to test the Function Block Program created. In continue to the steps in Creating New Function Block Program, proceed with the steps shown below.

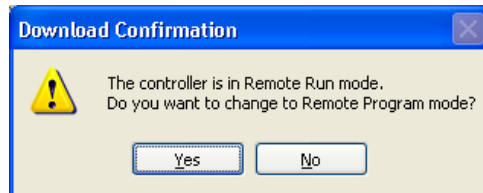
1. In the **Project Organizer**, right click on **Micro830**, and select **Download** to download the program:



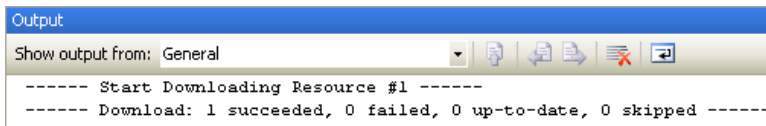
2. From the **Connection Browser**, select **2080-L30-16QWB**, and click on OK.



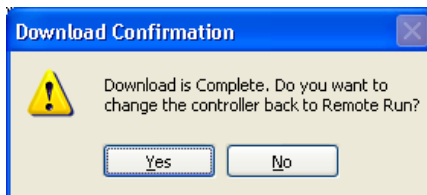
3. The following dialog box will appear for confirmation of the downloading if the controller is in RUN mode. Click on **Yes** to proceed.



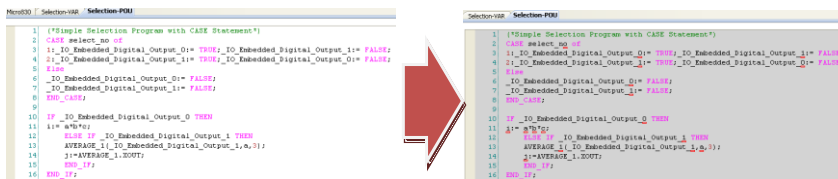
- In the completion of downloading the program, the **Output** window will display **Succeeded**



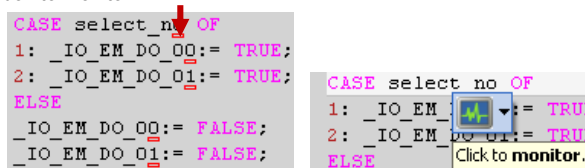
- The following window will appear to change from Program Mode to Run Mode. Click on **Yes** to proceed.



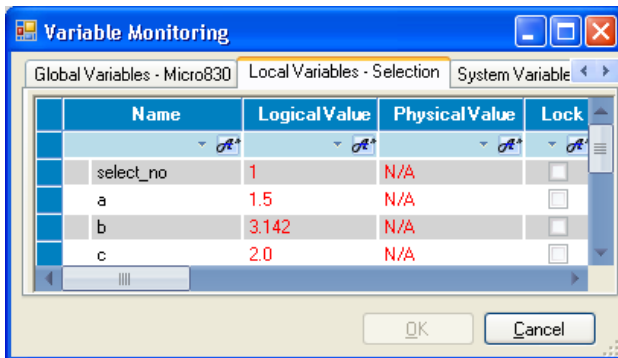
- Click on the at the Debug Toolbar, the programming workspace will change from white background to gray background.



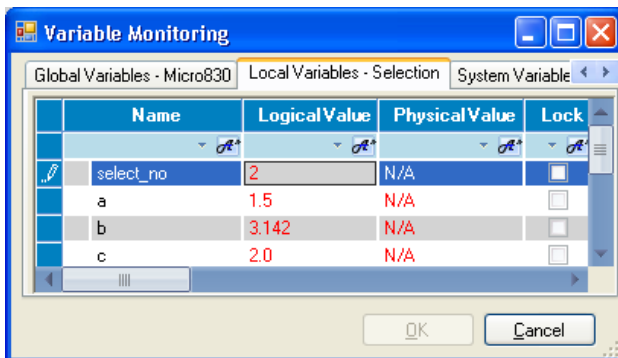
- To simulate the variable, run over the and the following popup dialog box will appear. Click on the dialog box to monitor.



8. The **Variable Monitoring** window will appear.



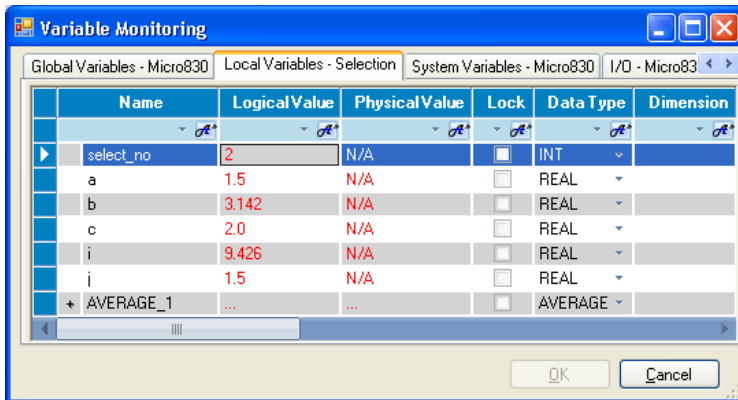
9. Change the value at the **Logical Value** of the variable `select_no`. to simulate the program.



Simulation for `select_no` using the demo kit output indicators:

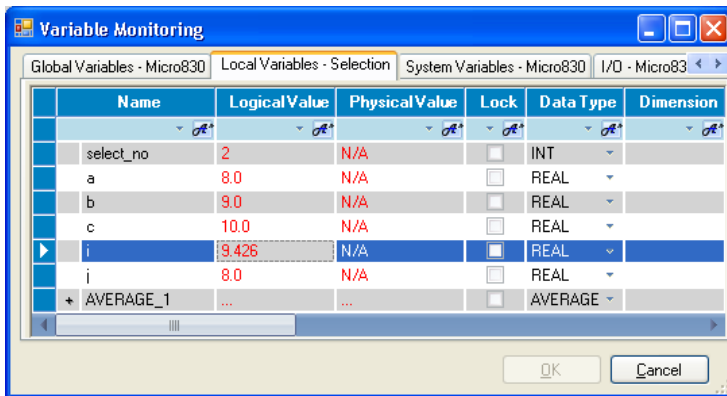
- In the demo kit, Output 0 should be lit when the `select_no` variable is 1. At the Logical Value of `select_no`. change to 2. Now, Output 0 should turn off, and Output 1 should lit.
- Change the value of `select_no` variable to 0 or 3, both Output 0 and Output 1 should turn off.
- The program logical is written so that if the value is not 1 or 2, both Output 0 and Output 1 should turn off.

10. To simulate the mathematic calculation, at the **Variable Monitoring** Window, change the value of a, b and c.

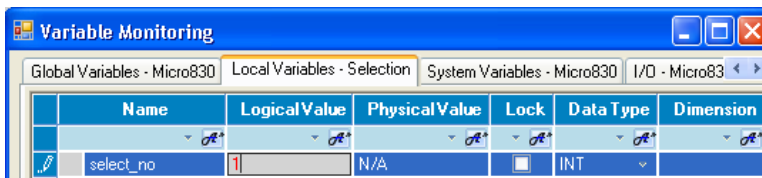


Simulation for the equation $i := a*b*c$;

Initial value of a is 1.5, b is 3.142 and c is 2.0, change the values to as shown below.



However, we expected i to equal 720.0. We need to change the value of the select_no to 1 to execute the equation $i := a*b*c$;



When the **select_no**'s value is changed to 1, the equation will be executed. The value will be shown in the **Variable Monitoring** window.


Name	Logical Value	Physical Value	Lock	Data Type	Dimension
select_no	1	N/A	<input type="checkbox"/>	INT	
a	8.0	N/A	<input type="checkbox"/>	REAL	
b	9.0	N/A	<input type="checkbox"/>	REAL	
c	10.0	N/A	<input type="checkbox"/>	REAL	
i	720.0	N/A	<input type="checkbox"/>	REAL	
l	8.0	N/A	<input type="checkbox"/>	REAL	
AVERAGE_1	<input type="checkbox"/>	AVERAGE	

The program is written in such:

IF _IO_EM_DO_00 THEN

$i := a*b*c;$

Therefore, only when the Output 0 = 1 will the equation be executed.

11. To stop the monitoring of the variable, click on  at the Debug Toolbar.
12. Then click on **Disconnect** to go offline, at the Micro830 tab.

