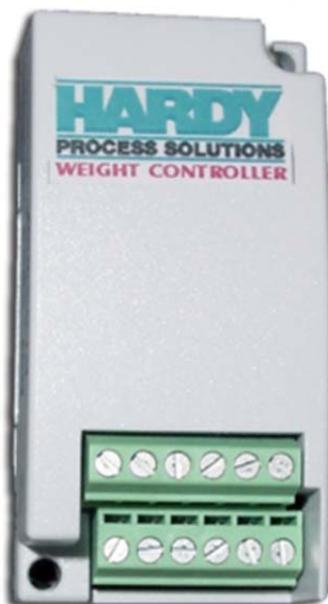


Hardy HI 2080

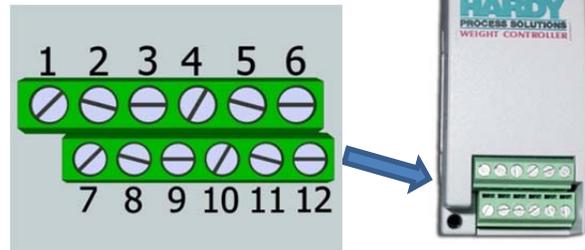
EASY 8 Quick Start & Programming Guide

- Section 1: Wiring Guide
Section 2: Function Block Description
Section 3: Using Function Blocks



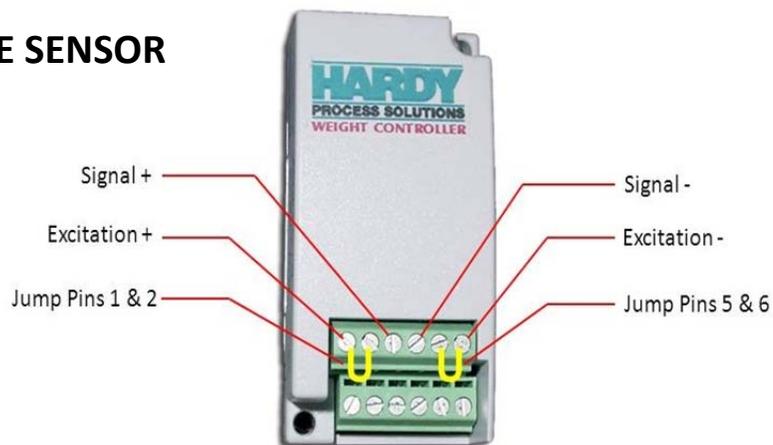
Section 1: Wiring Guide

| | | | |
|---|--------------|----|-----------------|
| 1 | Excitation + | 7 | No Connect |
| 2 | Sense + | 8 | No Connect |
| 3 | Signal + | 9 | Floating Ground |
| 4 | Signal - | 10 | Floating Ground |
| 5 | Sense - | 11 | Floating Ground |
| 6 | Excitation - | 12 | Floating Ground |



EXCITATION VOLTAGE: 5 VDC

4-WIRE SENSOR



Important: 4-wire sensor jump pins **1 & 2** and **5 & 6**

ATTENTION: Do not insert or remove module while the PLC is powered. "Hot-Swapping" may result in damage to the PLC and/or the module.

6-WIRE SENSOR



Section 2: Function Block Description

| NAME | FUNCTION |
|------------------------|--|
| Read_Wt_status | Provides weight and status reading from Plug In Module (PIM) |
| Zero | Sets Gross weight reading to zero |
| Tare | Sets Net weight reading to zero |
| Cal_Low | Sets low calibration point |
| Cal_High | Sets high calibration point |
| Set_Config | Writes configuration parameters to PIM |
| Read_Configuration | Read configuration parameters from PIM |
| Save | Saves configuration parameters to non-volatile memory |
| Set_Sensitivity_Filter | Sets gain and condition of filter in PIM |
| Set_defaults | Sets PIM back to default settings |
| Get_Cmd_Status | Read status of previous command |
| Get_Version | Reads PIM firmware version information |

Read_Wt_status: This block will read the weight, both gross and net, and the status of the module. This would be used constantly to update the weight reading and the status of the module. The motion status indicates if the weight reading is stable.

Zero: Function block used to zero the gross weight. Using this function block will force the gross weight to a zero value if the weight does not return to zero (for example when residual material is left in a vessel).

Tare: Function to zero the Net weight. Using this function block will cause the net weight to go to a value of zero. Typically used to eliminate the weight of containers or other material on the scale so that only the weight of the material is shown.

Cal_Low: Calibration low Command. This will set the low point in a two point calibration process. This would be used to normally set the low point at zero of a new calibration on the scale. This will use the Cal Low Weight as the reference for this low point and must match the amount of live load weight on the scale.

Cal_High: Calibration High Command. This will set the high point in a two point calibration. This would be used to set the high point of a new calibration. This will use the Span setting as the reference and must match the amount of live load weight placed on the scale for this command.

Set_Config: This will write the configuration parameters to the module. These parameters are:

NumAverages - the number of readings taken and averaged together to report as the weight;

Motion Tolerance - sets the amount of weight change over a one second period that determines the "stability" of the weight reading. A Motion indication means the weight is changing on the

scale, this motion status will prevent the unit from doing a zero command, tare command, cal low command or a cal high command.

Cal Low Weight - the reference for the amount of live load weight you have on the scale at the time of the Cal Low Command being run;

Cal High Weight - the reference for the amount of live load weight you have on the scale at the time of the Cal High Command being run.

Read_Configuration: Reads the configuration of the parameters established during the Set_Config and the Set_Sensitivity_Filter . This would be to check to see what configuration parameters are set in the PIM.

Save: Saves configuration setting to non-volatile memory. This should be used any time changes to the configuration or calibration are made to prevent loss of settings during a power-cycle.

Set_Sensitivity_Filter: Resets the gain of the A/D converter, sets filter and turns fir filter on/off. This adjusts the gain of the converter to get better resolution with different mv/v ratings of the load cells. Suggested Gain setting below.

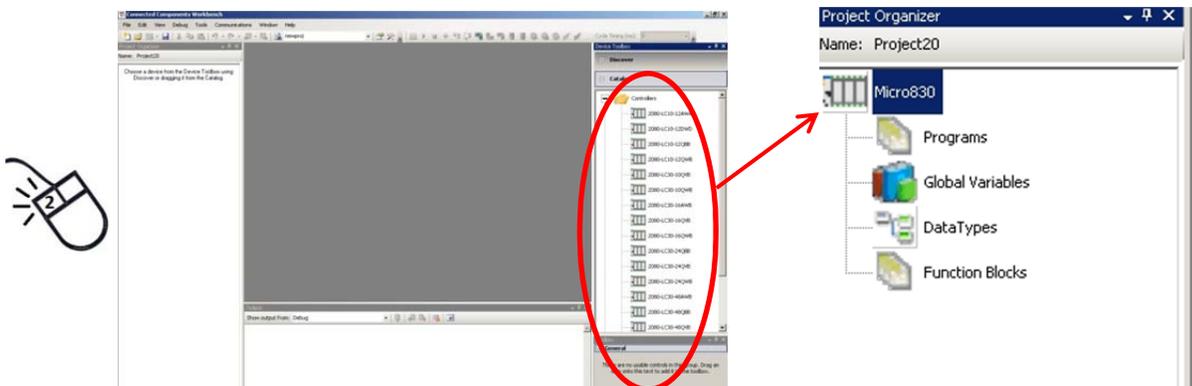
Get_Cmd_Status: Reads the status of the last command sent to the module.

Get_Version: Reads the version of the firmware in the module. Typically used in troubleshooting to provide module information during a Technical Support event.

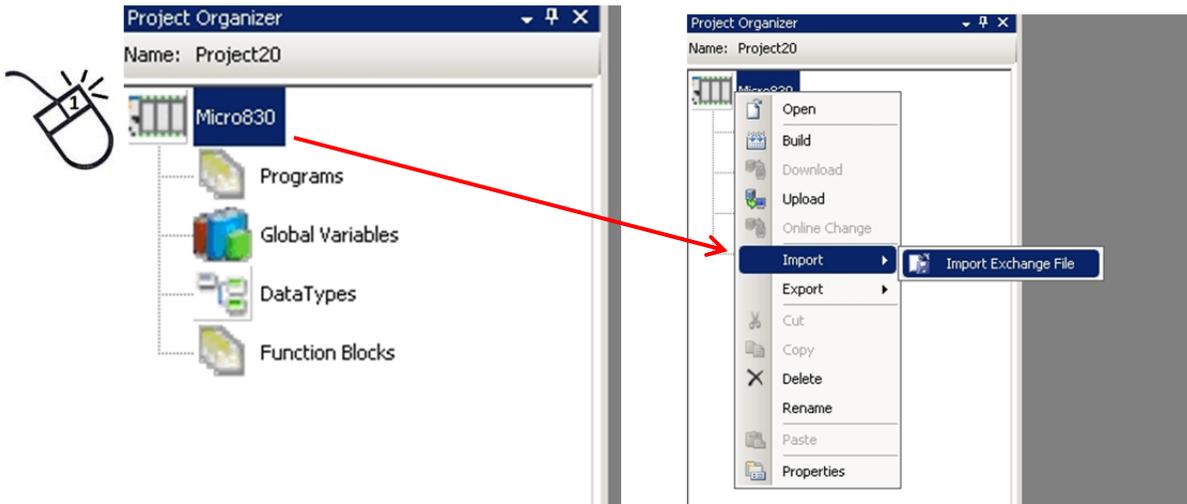
Set_Defaults: Sets the module back to factory default settings.

Section 3: Using Function Blocks

1. Start a new project with Connected Components Workbench software.
2. Choose your specific controller from a list on the right side of the screen by double clicking.



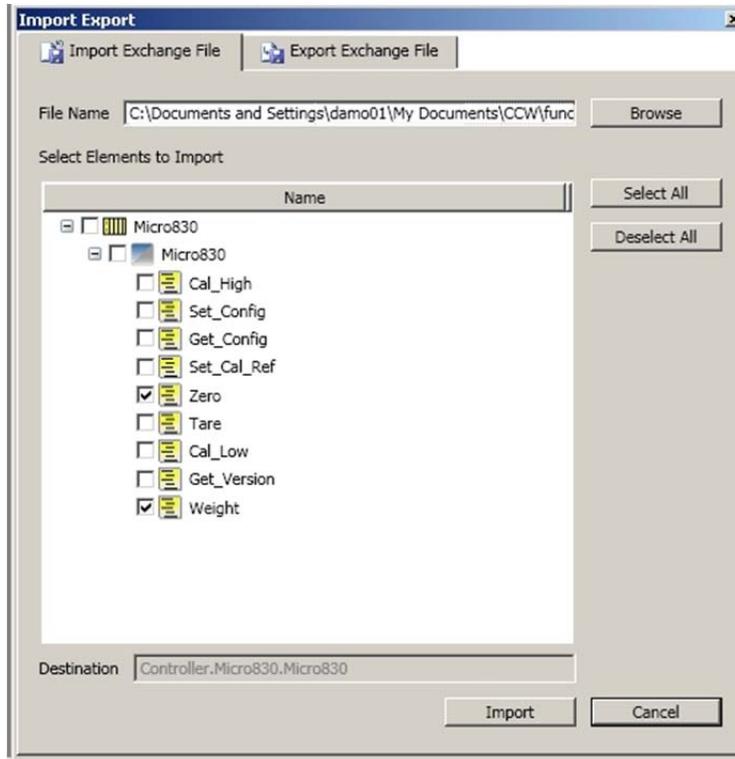
3. After double clicking your controller, a new project will open in the Project Organizer window.
4. Right click on the Micro830 and select to import an exchange file.



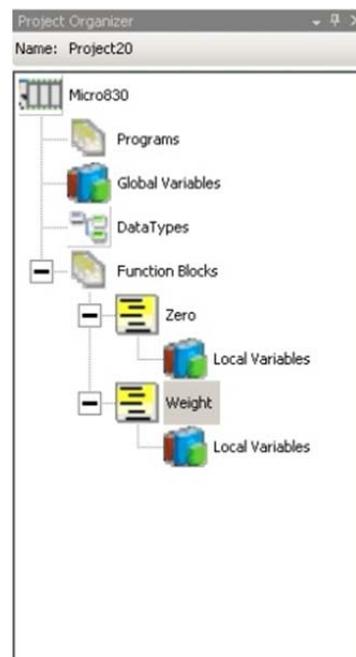
5. Select IMPORT EXCHANGE FILE and navigate to the EASY 8 FUNCTION BLOCK file.
(download: http://www.hardyinst.com/process_weighing/plc+plug-in+modules/easy+8%99+series+weigh+scale+module)
Select ALL or just the function blocks required for your project.

In the example below, Weight and Zero function blocks were selected.

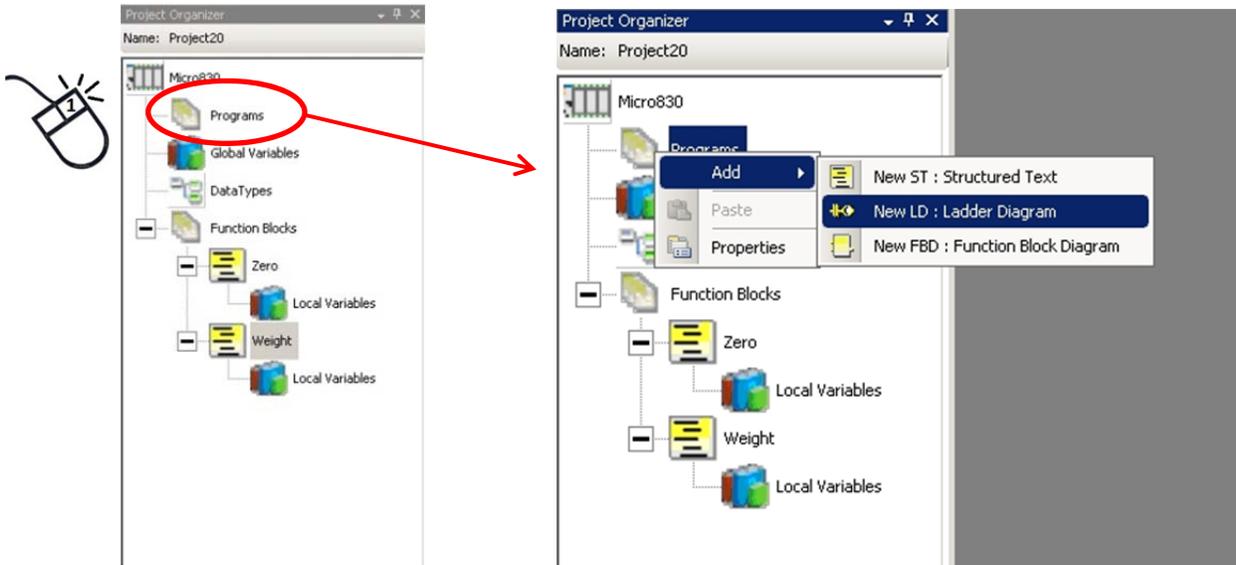
If you find you wish to add function blocks to a project, import them later in the same way.



6. After you have imported the function blocks, they will show up in the Project Organizer window as shown.

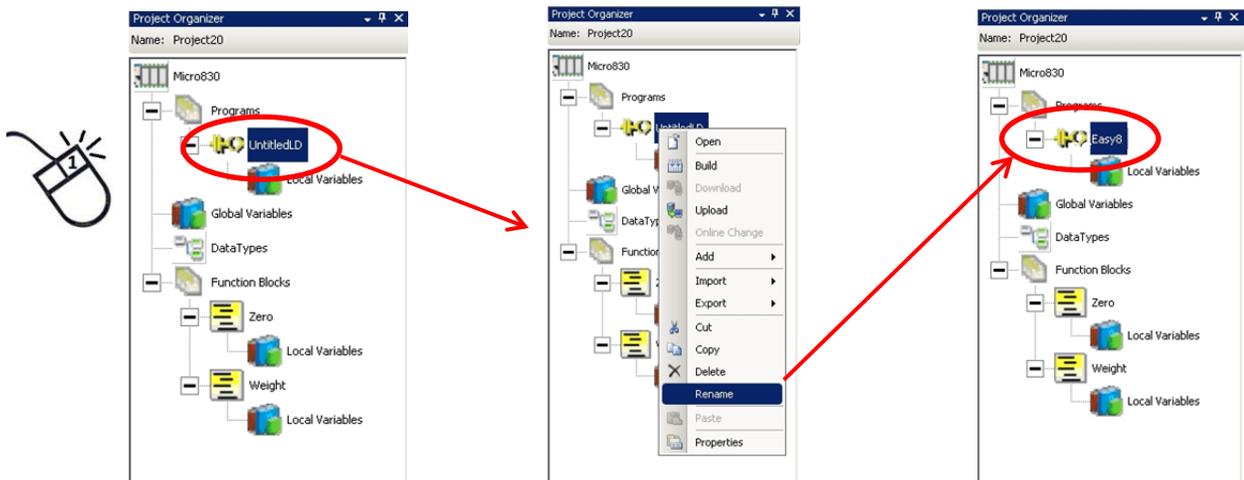


7. To create a ladder program using the function blocks, right click on "Programs" and add a new LD: Ladder Diagram.



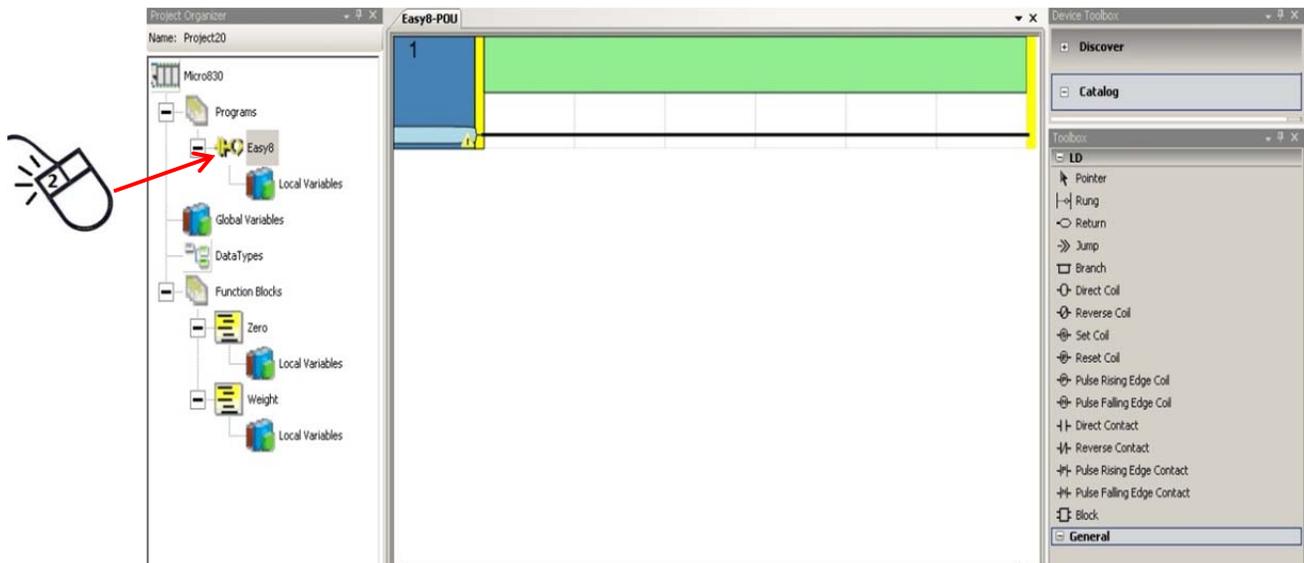
8. Right click on the UntitledLD program and select “Rename”.

In this example the new program has been called EASY8.



9. Double click on the program to open the ladder window.

The first rung of the new program will be shown with a list of instructions to the right.



For this example, the following selections are used:

Block –

Enables use of a function block in the logic.

Direct Contact –

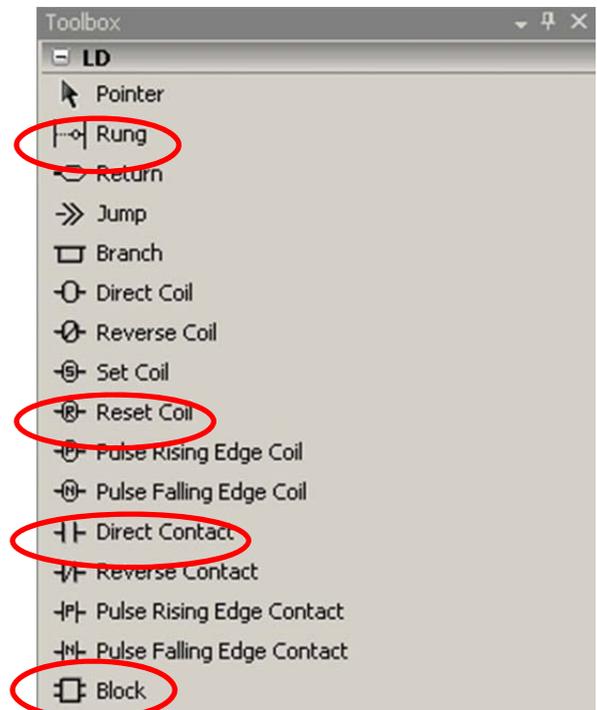
Enables verification of bit status (on or off).

Reset Coil –

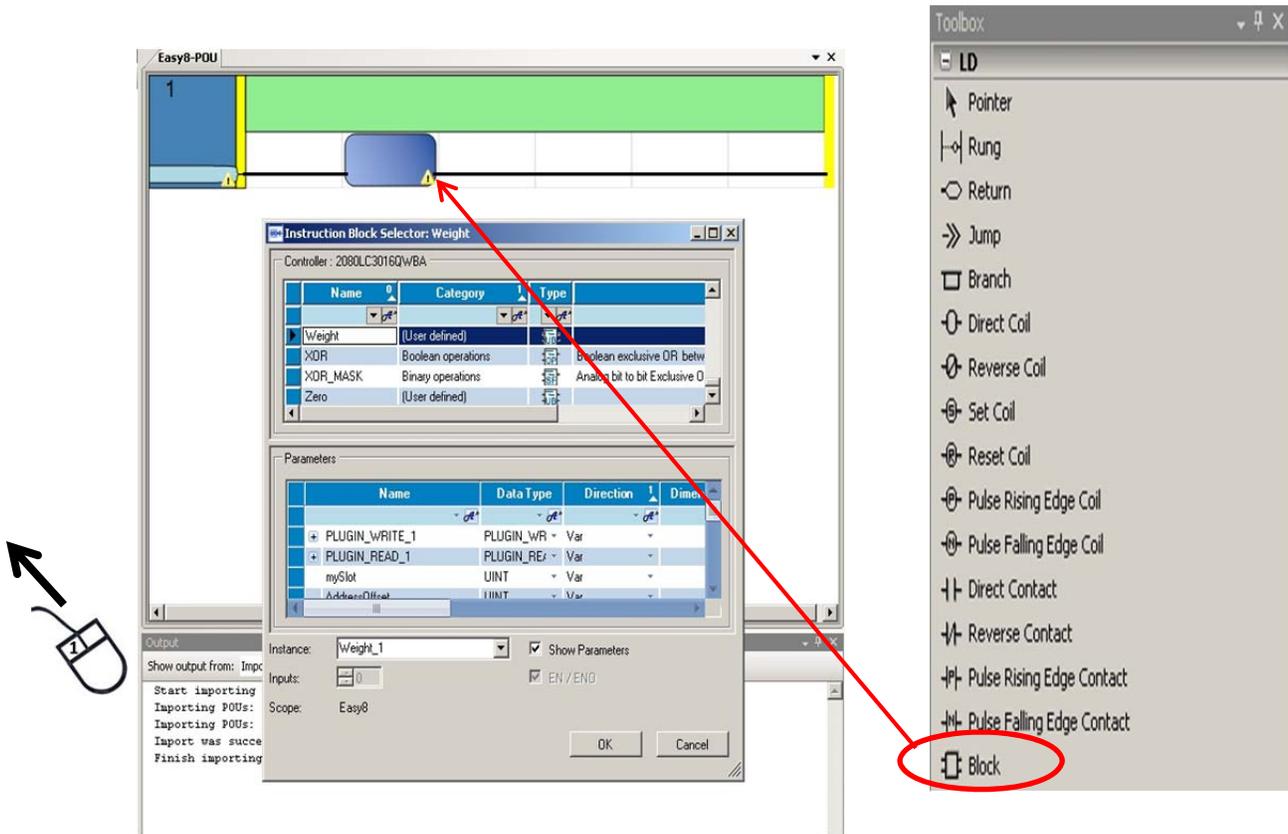
Enables change of a bit status from on to off.

Rung –

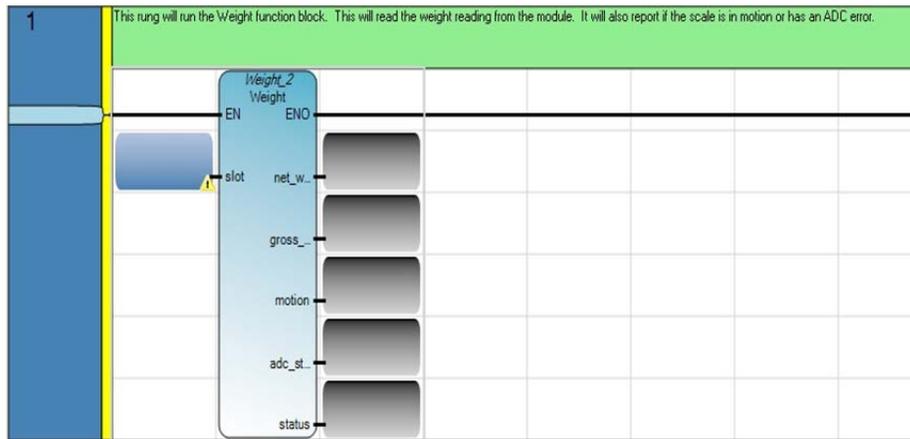
Adds another rung to the logic.



10. Click and drag a “block” to the rung and drop it. This will then bring up the Block selection box.

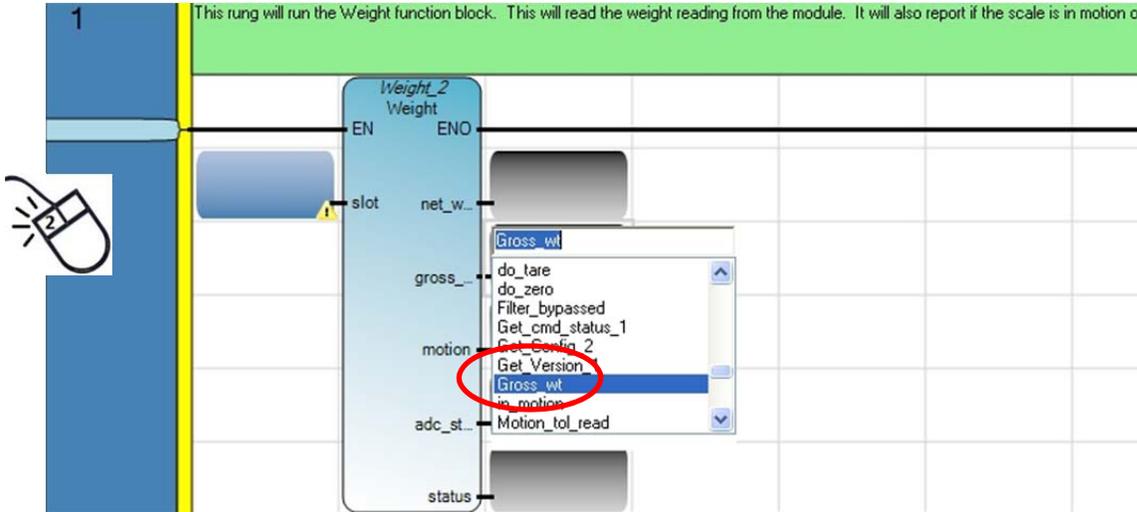


11. From the list by name, scroll down and select the function block to be inserted. In this example WEIGHT function block is used. Click Ok to insert it into the ladder diagram.



12. The WEIGHT function block has five outputs. A variable must be assigned to each output.

Assign a variable to each output by double clicking on the output box to open the Variable Selector window.

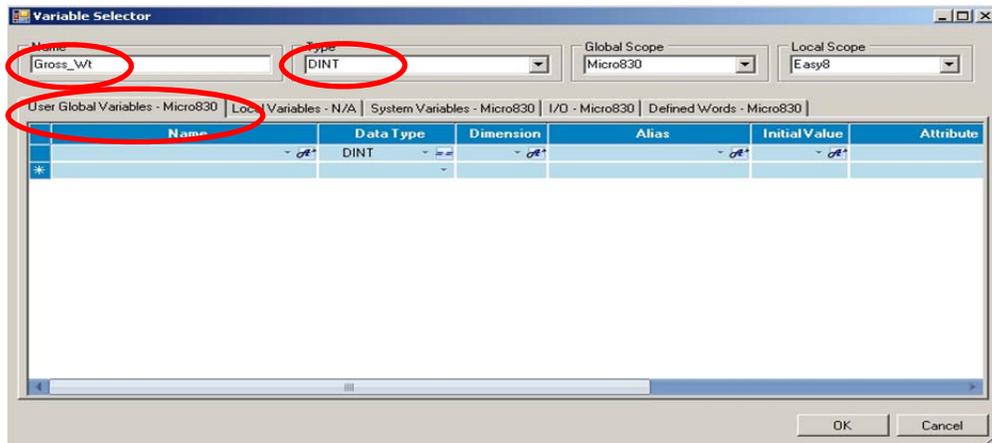


13. Click on the tab for the type of variable this will be, either Global or Local.
In this example GLOBAL variable was selected.

In the "Name" field, enter the name for this variable.
In this example, the name of this output is Gross_Wt.

In the "Type" field, select the type of register this will be.
In this example, the Gross weight will be a DINT (Double Integer) type.

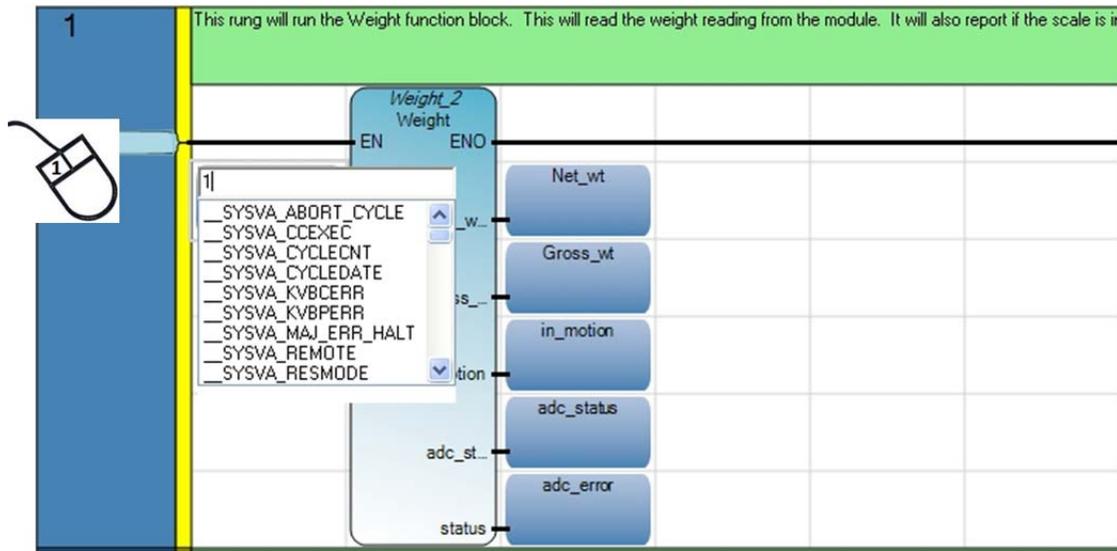
Click OK to accept.



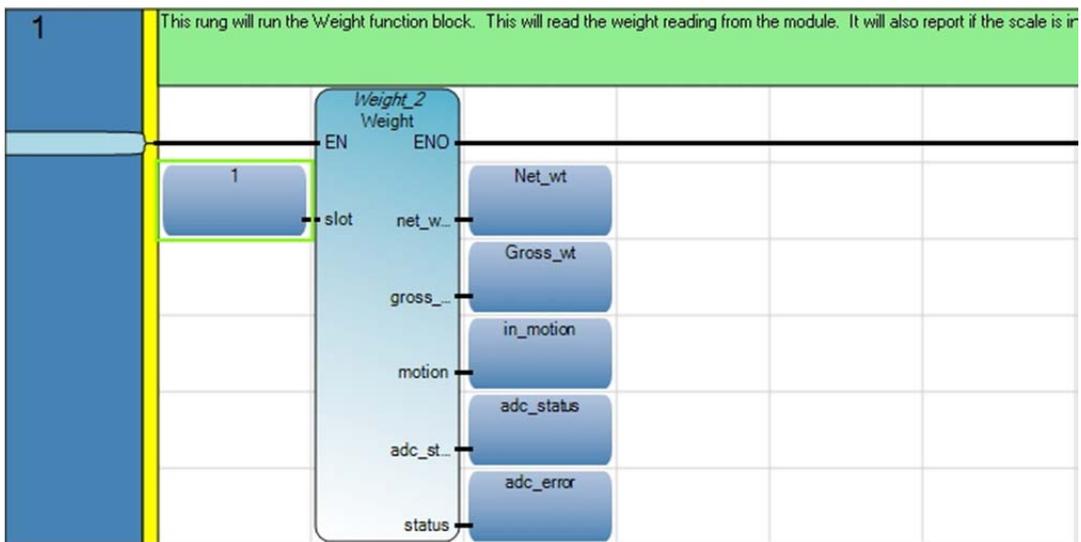
14. Repeat above to define the remaining outputs:

15. The WEIGHT function block also has an input for the slot number. The slot number is used to define the slot in the chassis to send the command to.

Single left click the slot output box to open the Variable Selector and define the slot variable. In this example, the module is in slot 1 and will be defined as a constant.



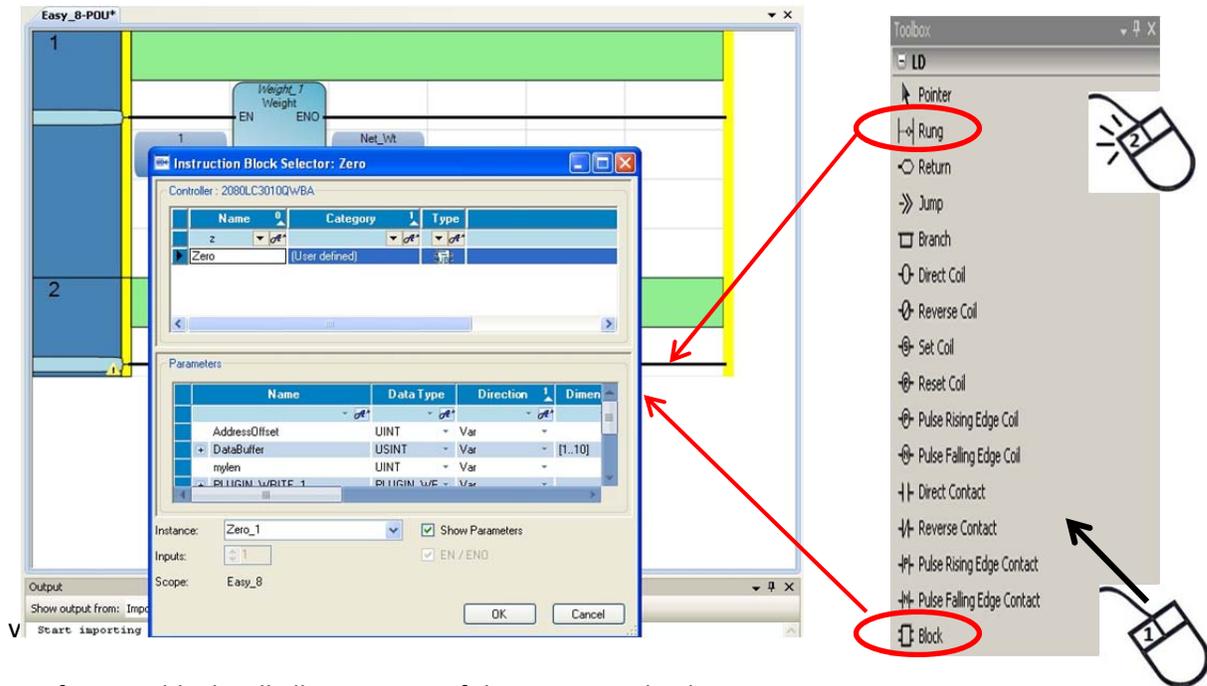
All the inputs and outputs for the Weight function block have now been defined.



In this example, the weight readings will be continuously updated, no further conditions will be placed on this rung.

16. Enter another rung by double-clicking on the “Rung” instruction in the toolbox list. This will add another rung in the diagram after the last rung.

17. Click and drag a “Block” instruction to this second rung and select the Zero function block from the list, click Ok to insert it into the ladder diagram.



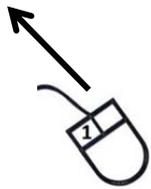
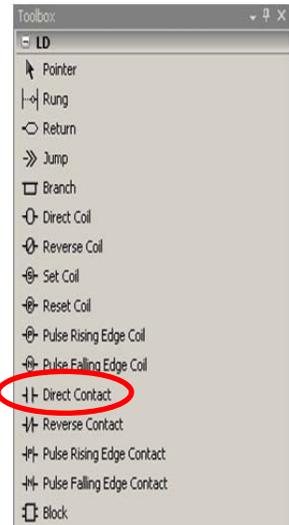
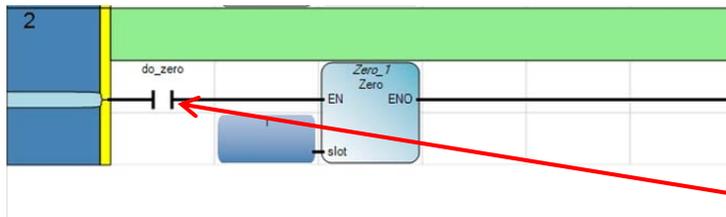
The Zero function block will allow zeroing of the gross weigh when run.

Set the slot number as before (in this example: 1).

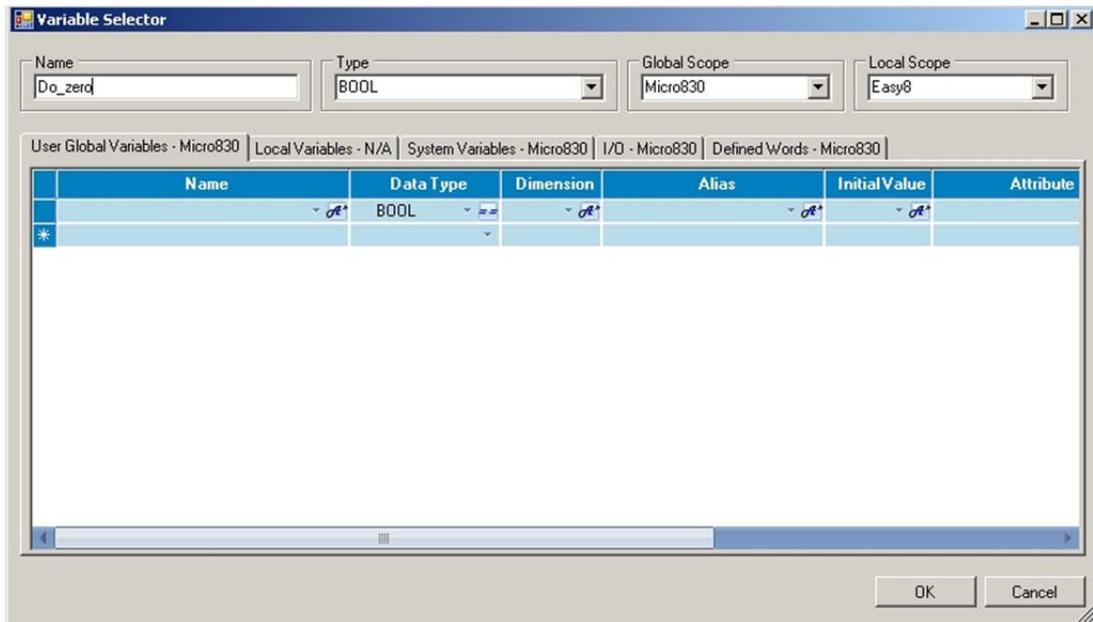


18. To prevent the scale from continuously zeroing, a condition will need to be placed on the rung.

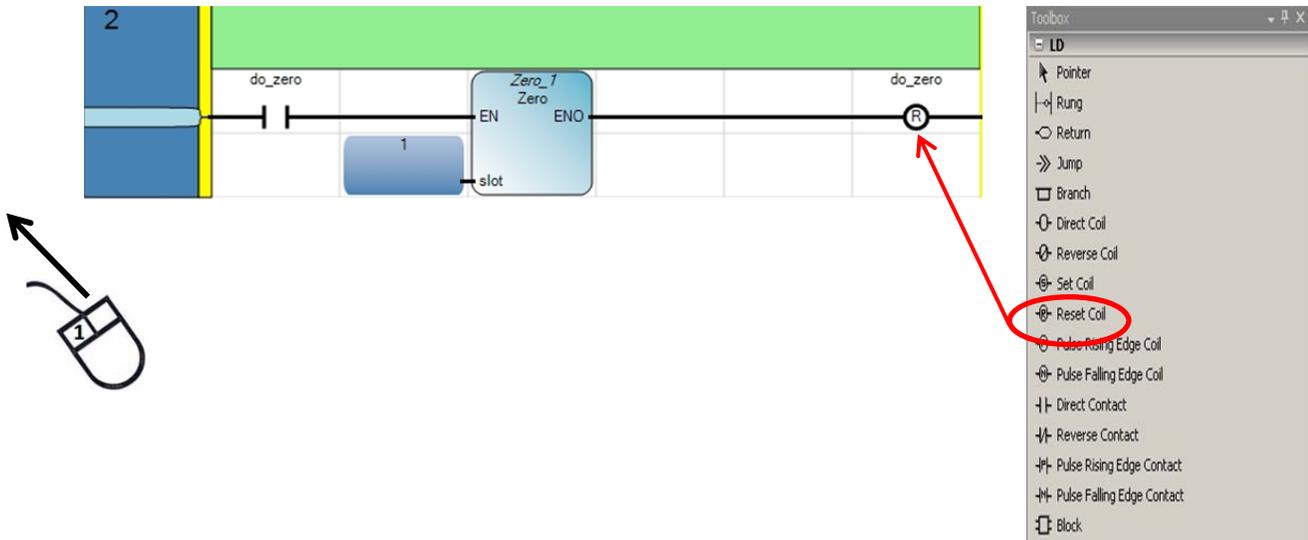
Add a Direct Contact from the toolbox by dragging and dropping onto the rung before (to the left of the function block).



In the Variable Selector box, make the Direct Contact command a Global variable, name it Do_zero and define it as a Boolean type variable.



- To run the Do_Zero command to only once, the Do_zero bit will need to be turned off after the function block is run. This is accomplished by using a Reset Coil command from the toolbox. Drag and drop it at the end of the Zero rung and name it Do_Zero when the variable selector box pops up.



20. Once the ladder rungs are completed, the project will need to be compiled to check for any errors and prepare it to be uploaded into the PLC.

Right click on the Micro830 and select "Build".

The screenshot shows two windows from a software interface. On the left is the 'Project Organizer' window, showing a tree view with 'Micro830' selected. A context menu is open over 'Micro830', with 'Build' highlighted. A red arrow points from 'Build' to the 'Output' window on the right. The 'Output' window displays the following text:

```

Download from: Build
----- Build started: Project: Controller, Configuration: 2006G2014Q0A -----
----- Clean started: Project: Controller -----
!!! file(s) deleted -----
----- Clean end -----
----- Build started: Project: Controller -----
Checking database...
----- Pre-build project: CONTROLLER -----
----- Pre-build resource: MICRO830 Configuration: MICRO830 -----
Pre-compiling for 2006G2014Q0A
ZERO
WEIGHT
EAT7F
MICRO830: 0 error(s), 0 warning(s)
CONTROLLER: 0 error(s), 0 warning(s)
Checking database...
----- Build project: CONTROLLER -----
----- Build resource: MICRO830 Configuration: MICRO830 -----
Compiling for 2006G2014Q0A
ZERO
WEIGHT
EAT7F
Linking for 2006G2014Q0A
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 *****

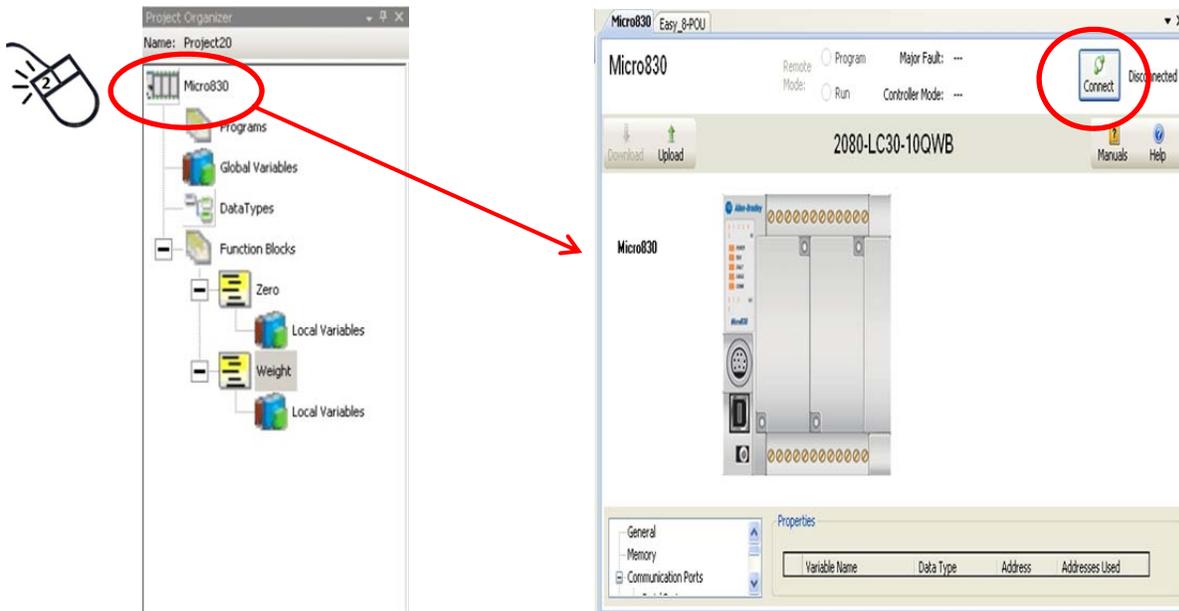
```

Once the project has gone through the build process, the results will be displayed in the Output window. Here is an example with Output window showing the build was successful.

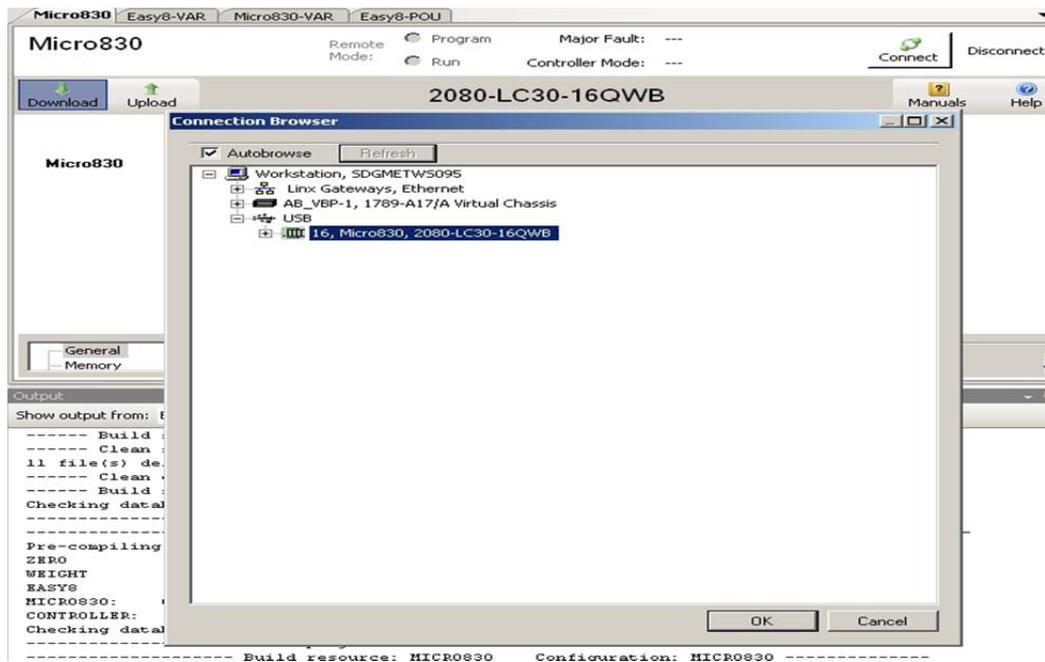
```
Output
Show output from: Build
----- Build started: Project: Controller, Configuration: 2080LC3016QWBA -----
----- Clean started: Project: Controller -----
11 file(s) deleted
----- Clean end -----
----- Build started: Project : Controller -----
Checking database...
----- Pre-build project: CONTROLLER -----
----- Pre-build resource: MICRO830 Configuration: MICRO830 -----
Pre-compiling for 2080LC3016QWBA
ZERO
WEIGHT
EASYS
MICRO830: 0 error(s), 0 warning(s)
CONTROLLER: 0 error(s), 0 warning(s)
Checking database...
----- Build project: CONTROLLER -----
----- Build resource: MICRO830 Configuration: MICRO830 -----
Compiling for 2080LC3016QWBA
ZERO
WEIGHT
EASYS
Linking for 2080LC3016QWBA
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 =====
```

After the build process, the project must be transferred to the PLC.

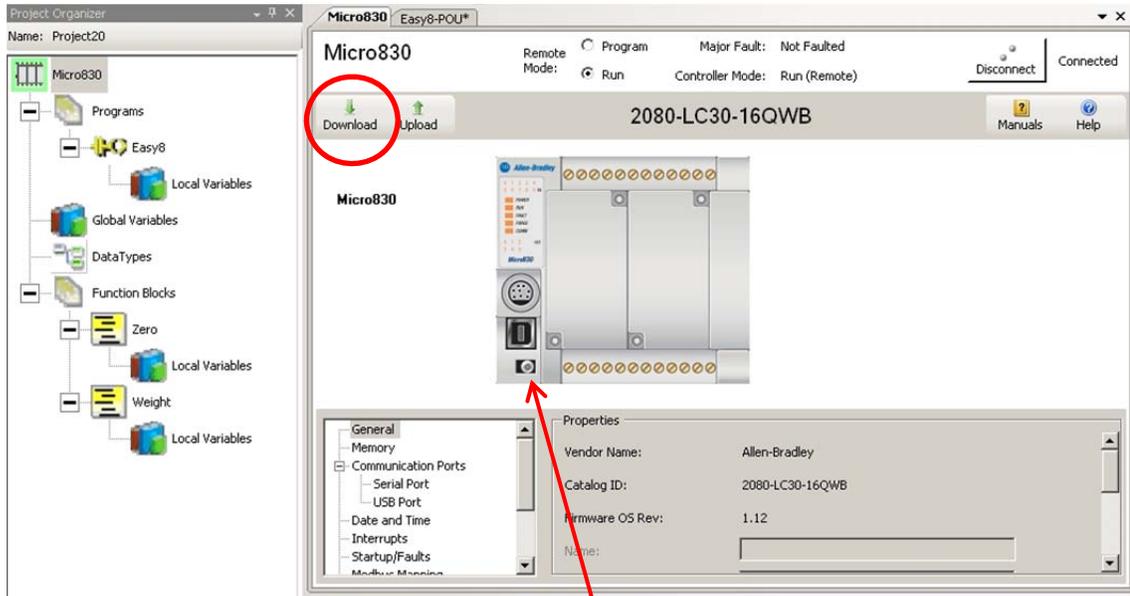
Double click on the Micro830 in the project organizer window to bring up the Micro830 configuration window.



Click on the Connect Button and select the communications path to the processor. In this example the USB port connection was used, click OK.

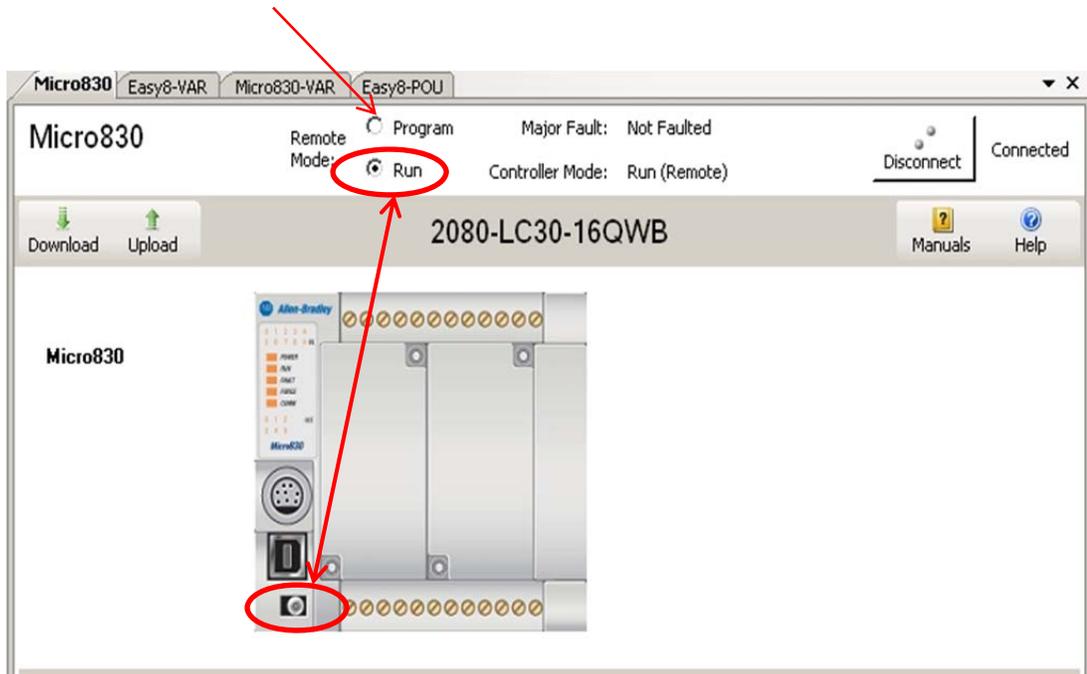


21. Once a connection has been established, download the program to the PLC by clicking on the Download button as shown.

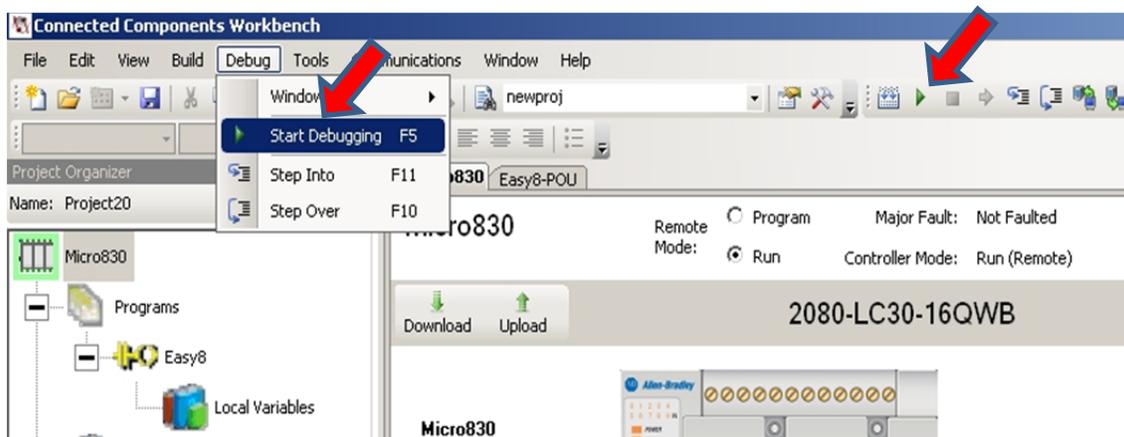


NOTE: Depending on the position of the **"RUN-REM-PROG"** switch on the PLC, it may necessary move the switch to either the program or the remote position to complete the download.

22. Once the download is completed, activate the run mode by either switching the selector switch on the processor, or if the switch in the "rem" position, click on the Run selection.



23. To monitor the process, run Debugging:
- by pressing F5 key
 - by clicking on Debug and clicking on “Start Debugging”
 - clicking on the green arrow icon shown below.



24. In the debug mode, on the program diagram window, right click on any of the variables and to view and change the values or conditions of the variables.

The screenshot shows the Micro830 Easy8-POU interface. Rung 1 is highlighted in red, indicating it is active. Rung 2 is highlighted in blue, indicating it is inactive. The Variable Monitoring window is open, showing the following data:

| Name | Logical Value | Physical Value | Lock | Data Type | Dimension | Alias | Comment |
|-----------|---------------|----------------|------|-----------|-----------|-------|---------|
| Net_Wt | 5 | N/A | | DINT | | | |
| Gross_Wt | 5 | N/A | | DINT | | | |
| in_motion | | N/A | | BOOL | | | |
| do_zero | | N/A | | BOOL | | | |

NOTE: If a rung is red then it is in a True state, if it is blue then it is a False state. In the example below the first rung, which is running continuously is red and the second rung, which is conditioned with a trigger bit before it can run is blue.

25. In the example below, 5lbs have been placed on the scale.

The screenshot shows the Micro830 Easy8-POU interface. Rung 1 is highlighted in red, indicating it is active. Rung 2 is highlighted in blue, indicating it is inactive. The Variable Monitoring window is open, showing the following data:

| Name | Logical Value | Physical Value | Lock | Data Type | Dimension | Alias | Comment |
|-----------|---------------|----------------|------|-----------|-----------|-------|---------|
| Net_Wt | 5 | N/A | | DINT | | | |
| Gross_Wt | 5 | N/A | | DINT | | | |
| in_motion | | N/A | | BOOL | | | |
| do_zero | | N/A | | BOOL | | | |

26. In the Variable Monitoring window, click on the box for Do_zero logical value to change it from low to high. The result of turning on the Do_zero bit will cause the gross weight to be zero'd. Notice in the example below that the gross weight reading has gone to zero.

| Name | Logical Value | Physical Value | Lock | Data Type | Dimension | Alias | Comment |
|-----------|-------------------------------------|----------------|--------------------------|-----------|-----------|-------|---------|
| Net_Wt | 0 | N/A | <input type="checkbox"/> | DINT | - | | |
| Gross_Wt | 0 | N/A | <input type="checkbox"/> | DINT | - | | |
| in_motion | <input type="checkbox"/> | N/A | <input type="checkbox"/> | BOOL | - | | |
| do_zero | <input checked="" type="checkbox"/> | N/A | <input type="checkbox"/> | BOOL | - | | |

Ongoing Product Support

Our Hardy Service Center is customer-focused and provides solutions that are valued by our customers. We offer a unique blend of support options that gives you the ability to maintain your facility your way. We are ready to join your team whenever and wherever you need us. Contact us at: hardysupport@hardysolutions.com or (800) 821-5831 Option 4 (858-292-2710 Option 4 outside the U.S.A)