

Connected Components Workbench Task-based Quick Start

QS# GS04 – Creating a New Ladder Diagram 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 51
- 2080-LC30-16QWB v1.03

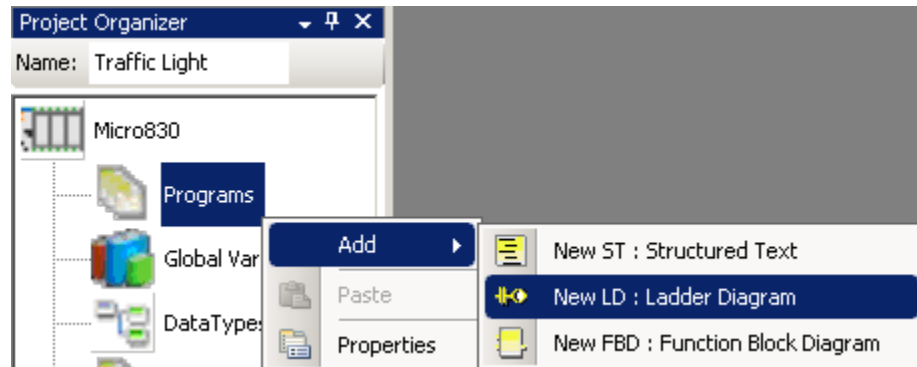
Prerequisite

- QS# GS03 – Creating a User Defined Function Block

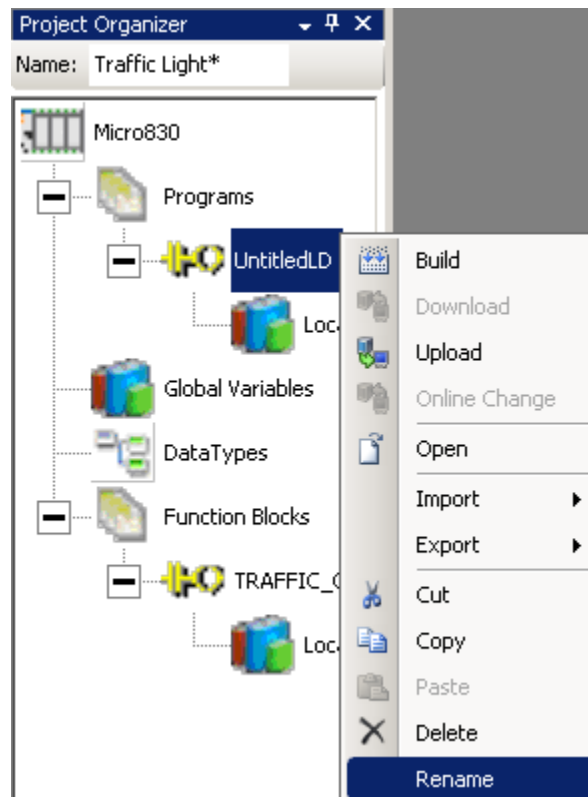
Creating a New Ladder Diagram Program

This quick start will show you how to create a new Ladder Diagram Program that uses the **TRAFFIC_CONTROLLER_FB** User Defined Function Block.

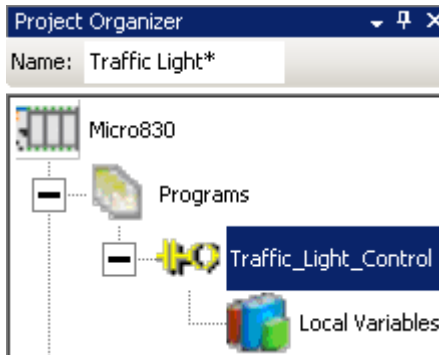
1. Under **Project Organizer**, right click on **Programs** select **Add** and select **New LD : Ladder Diagram** **Diagram**:



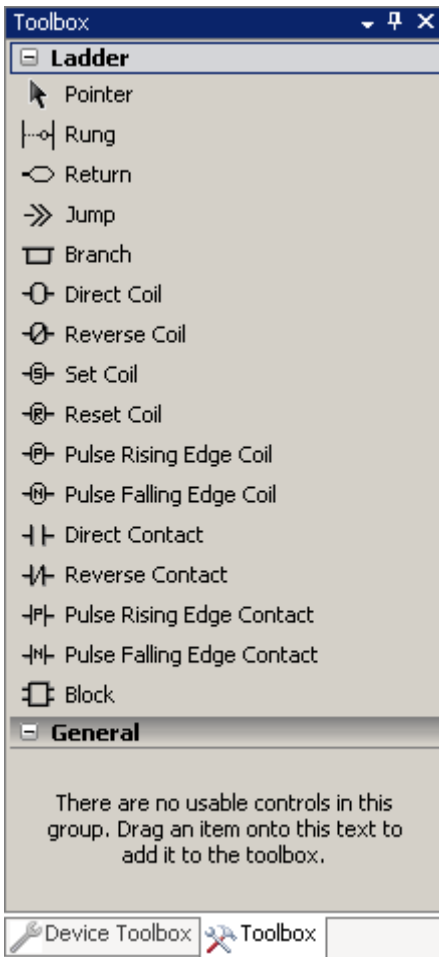
2. Right click on **UntitledLD** and select **Rename**:



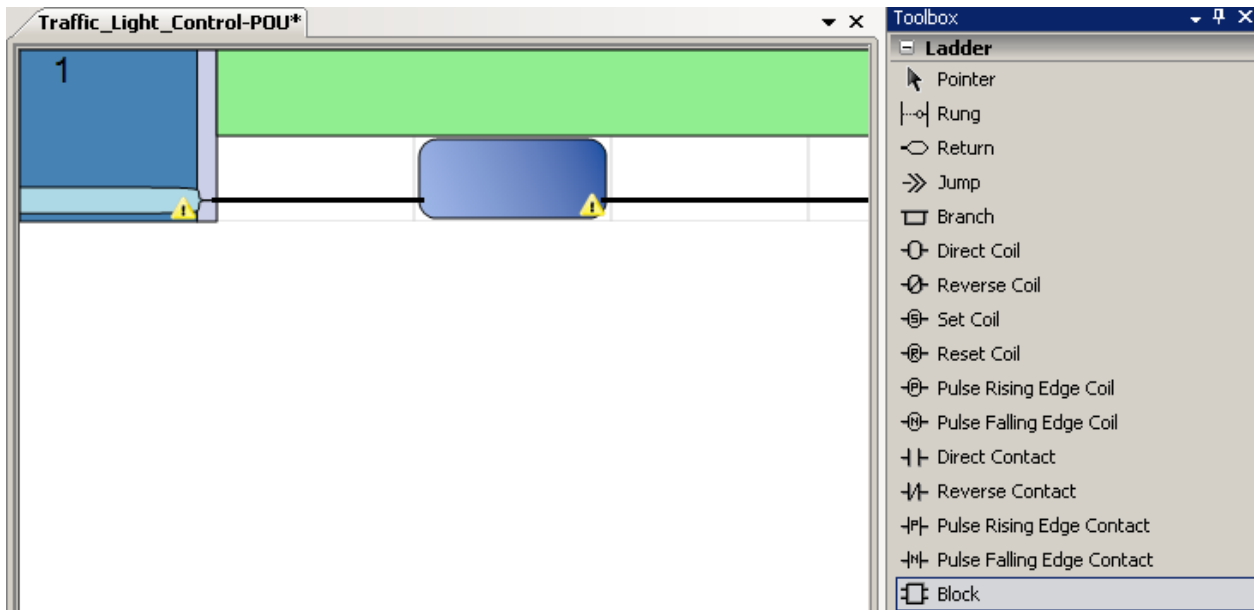
3. Type **Traffic_Light_Control** and Enter:



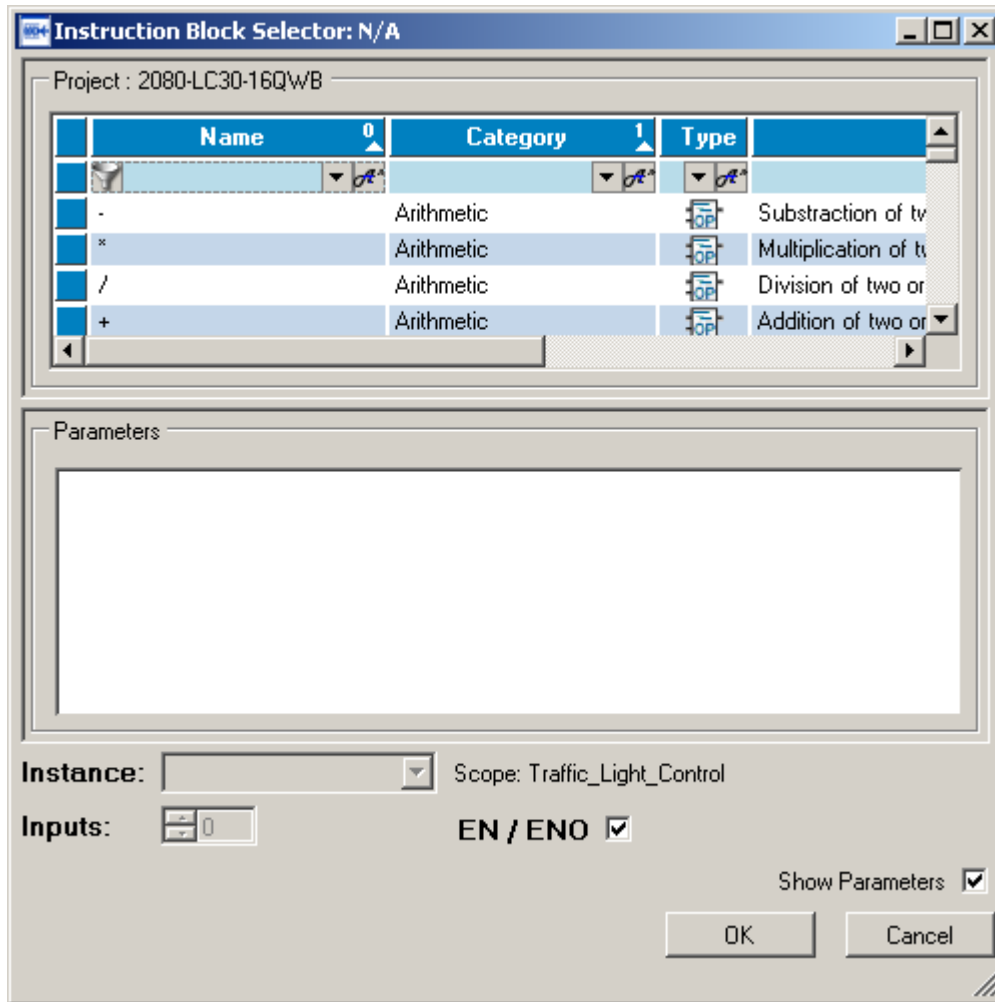
4. Double click on **Traffic_Light_Control** within **Project Organizer** to start editing the ladder logic program.
5. Click on the **Toolbox** tab in the lower right-hand corner and click on the **+** in front of **Ladder** to list the available ladder instructions:



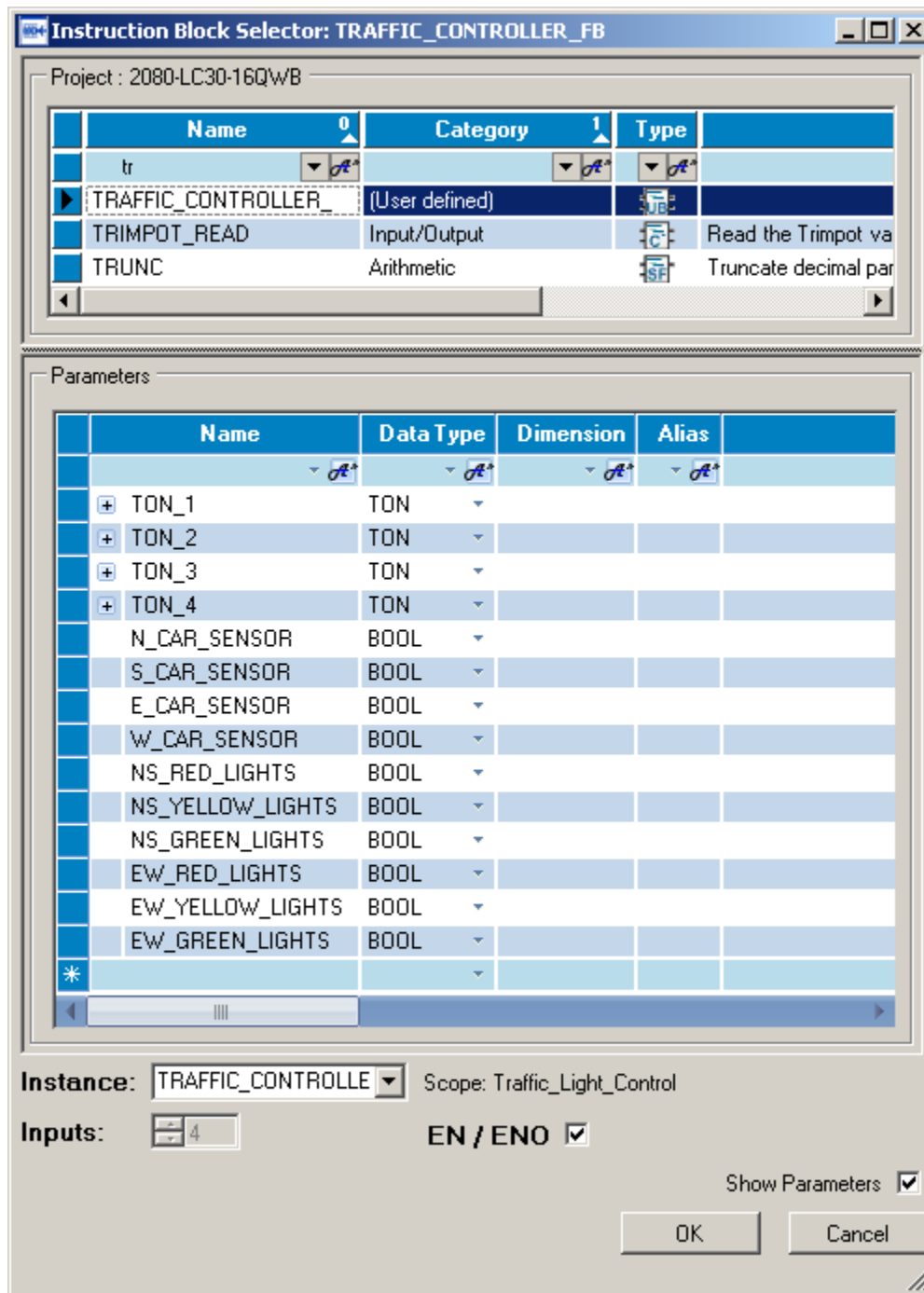
6. Click and hold on the **Block** instruction within the **Toolbox** and drag a Block onto the rung as shown:



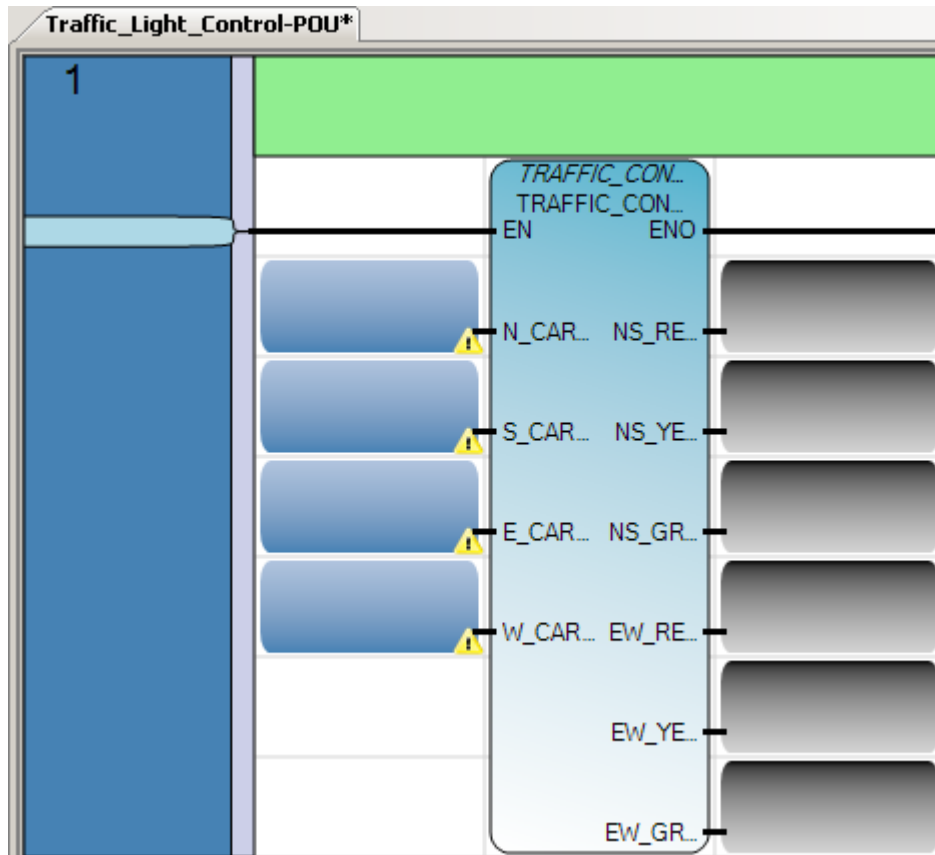
7. When you release the mouse button, the **Instruction Block Selector** screen appears:



- Under **Name**, type **tr** and note that only the instructions starting with tr are listed. Click on **TRAFFIC_CONTROLLER_FB** and notice that all of the **Parameters** associated with this function block are listed below it:



9. Make sure that EN/ENO is checked and then click **OK**. The Traffic Controller function block should be displayed on the rung as shown below:



10. By convention, function blocks list inputs on the left-hand side of the block and outputs on the right-hand side of the block. In order to see the full names and data types of the variables that these inputs and outputs are associated with, move your cursor over the block - you should get the following listing:

```

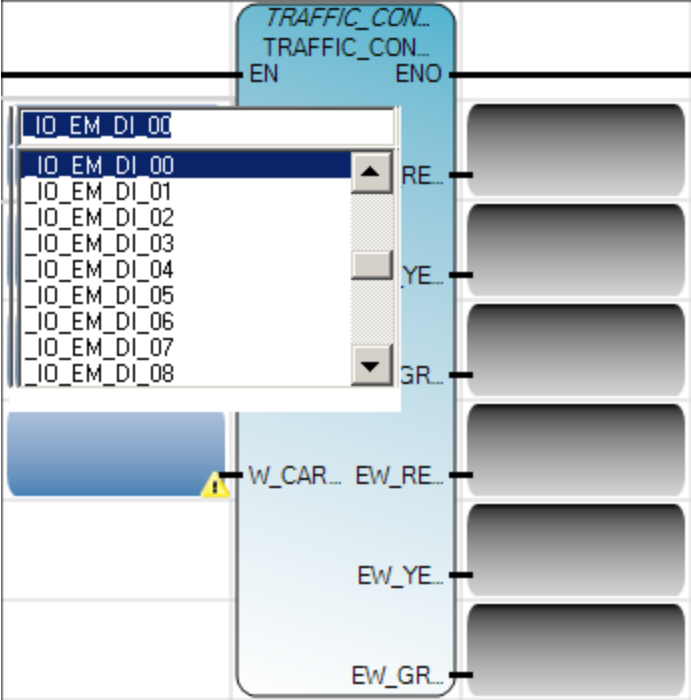
TRAFFIC_CONTROLLER_FB
TRAFFIC_CONTROLLER_FB_1

Inputs
-----
EN:
N_CAR_SENSOR: Bool
S_CAR_SENSOR: Bool
E_CAR_SENSOR: Bool
W_CAR_SENSOR: Bool

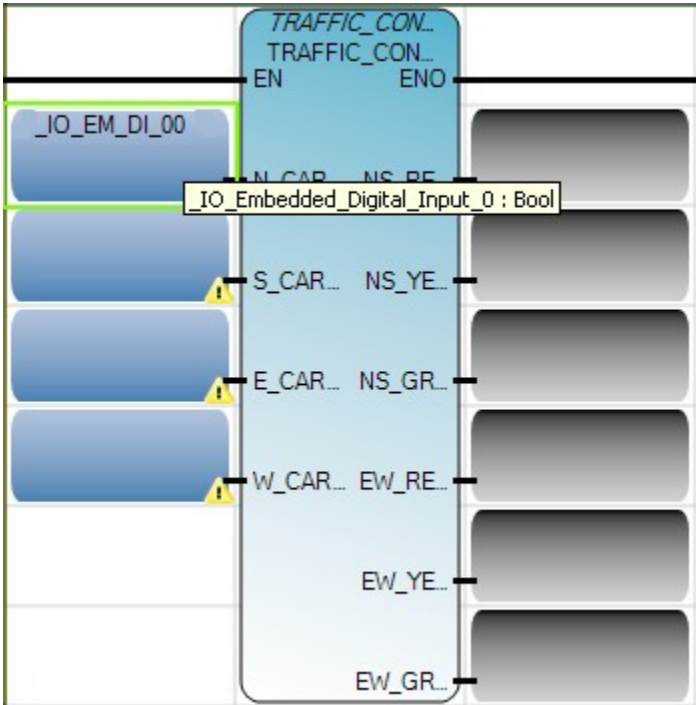
Outputs
-----
ENO:
NS_RED_LIGHTS: Bool
NS_YELLOW_LIGHTS: Bool
NS_GREEN_LIGHTS: Bool
EW_RED_LIGHTS: Bool
EW_YELLOW_LIGHTS: Bool
EW_GREEN_LIGHTS: Bool

```

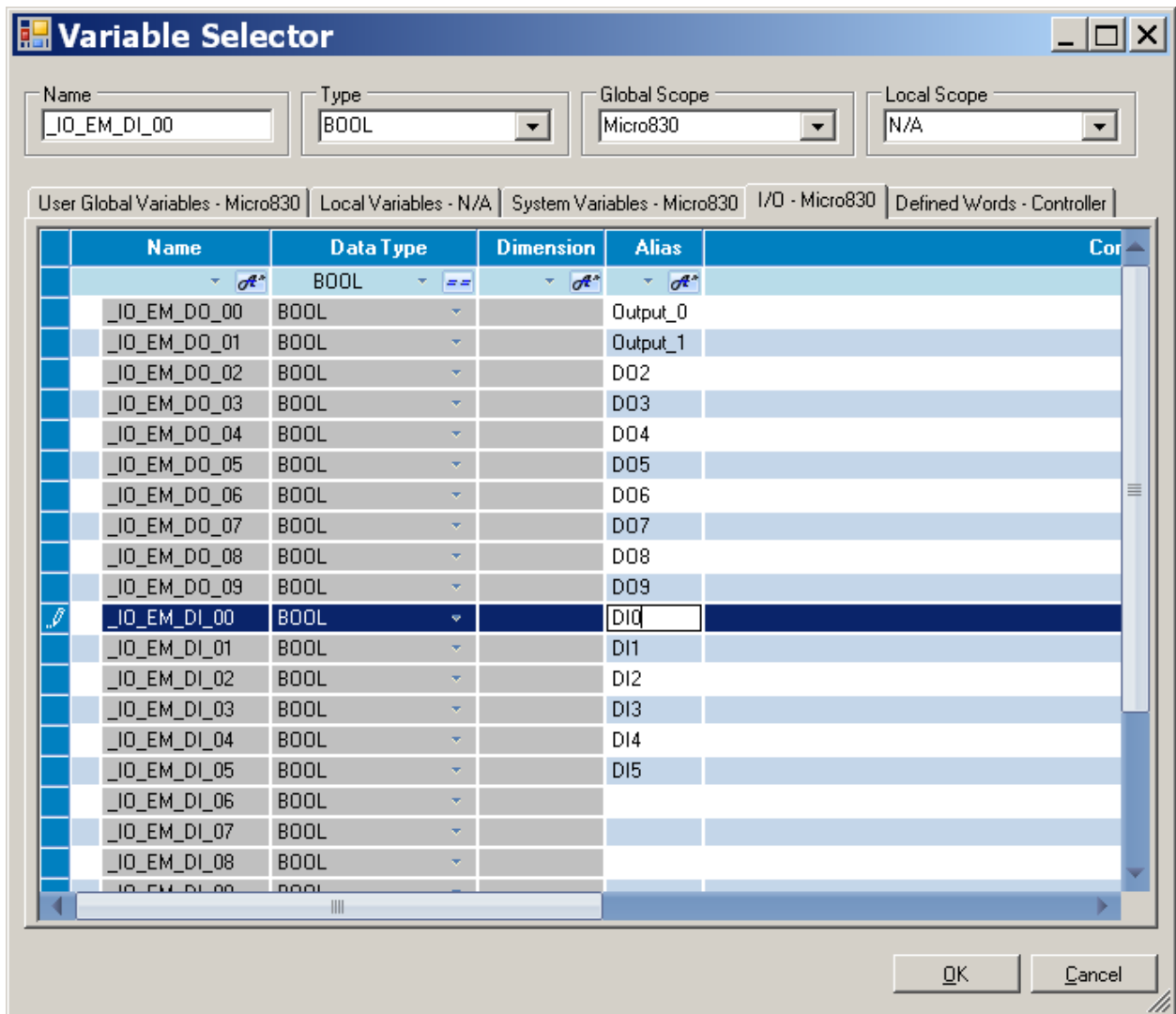
- 11. The first function block input that connects directly to the ladder rung is the function block enable (EN) bit. The remaining four function block inputs are “real world” inputs that indicate whether a car is waiting at a red light in any of the four possible directions – North, South, East and West. These inputs get mapped to four Boolean input variables local to the function block: **N_CAR_SENSOR**, **S_CAR_SENSOR**, **E_CAR_SENSOR** and **W_CAR_SENSOR**. You are going to assign four Micro830 controller inputs to these function block inputs.
- 12. Click on the top of the input variable block that connects to **N_CAR...** and you will get a dropdown menu of all the existing variable names that could be assigned to **N_CAR_SENSOR**. Scroll down and select **_IO_EM_DI_00** and enter:



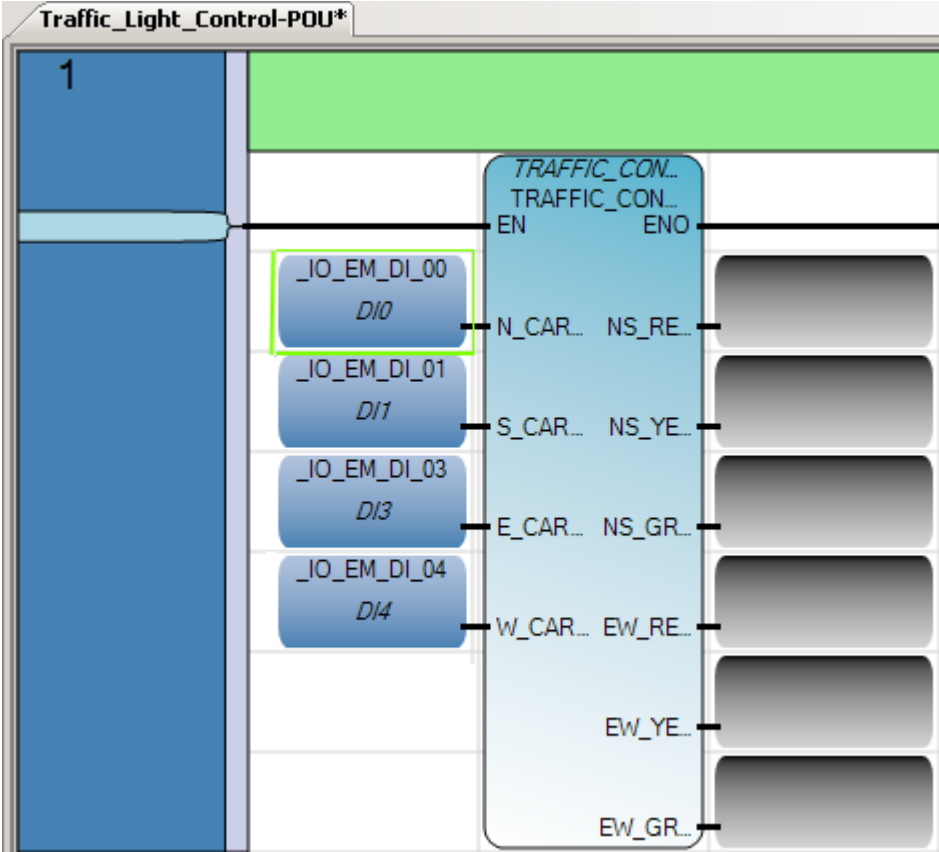
13. Notice that because of the length of embedded input 0 variable name, it is hard to tell what input is actually assigned to **N_CAR_SENSOR** when viewing the function block. One way to tell is to position your cursor over the block as shown below.



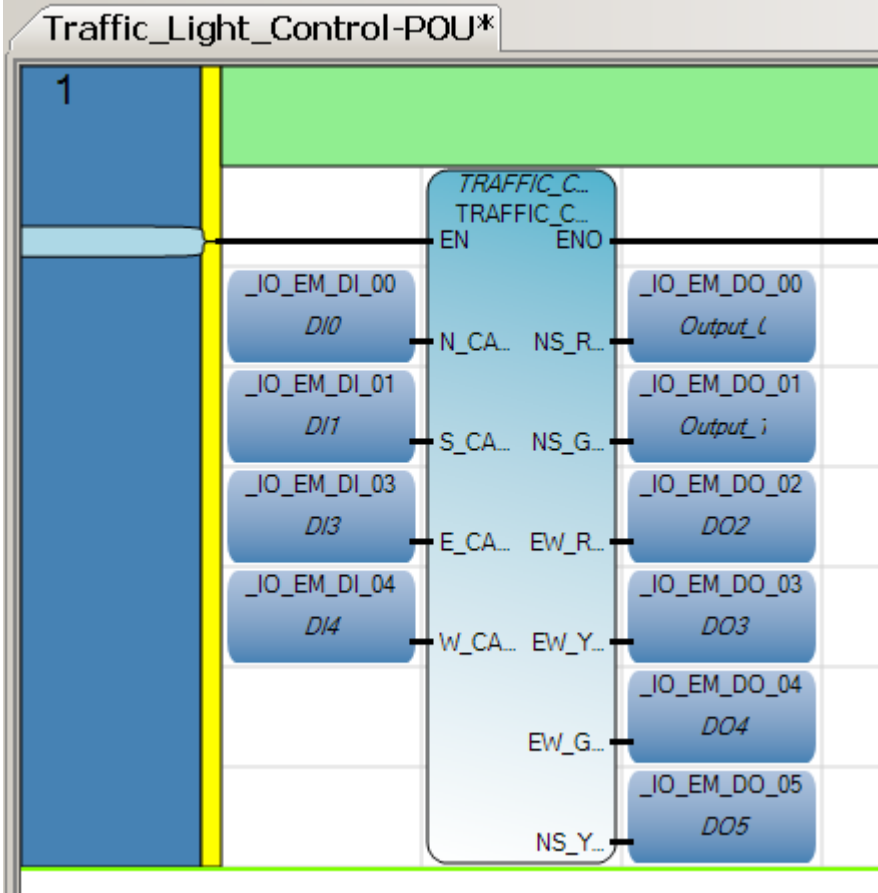
14. Another way is to assign shorter Alias names to these variables. Double click on the first input block – this brings up the **Variable Selector** screen. Go ahead and type in Alias names for the six outputs (**DO0-DO5**) and the first six inputs (**DI0-DI5**):



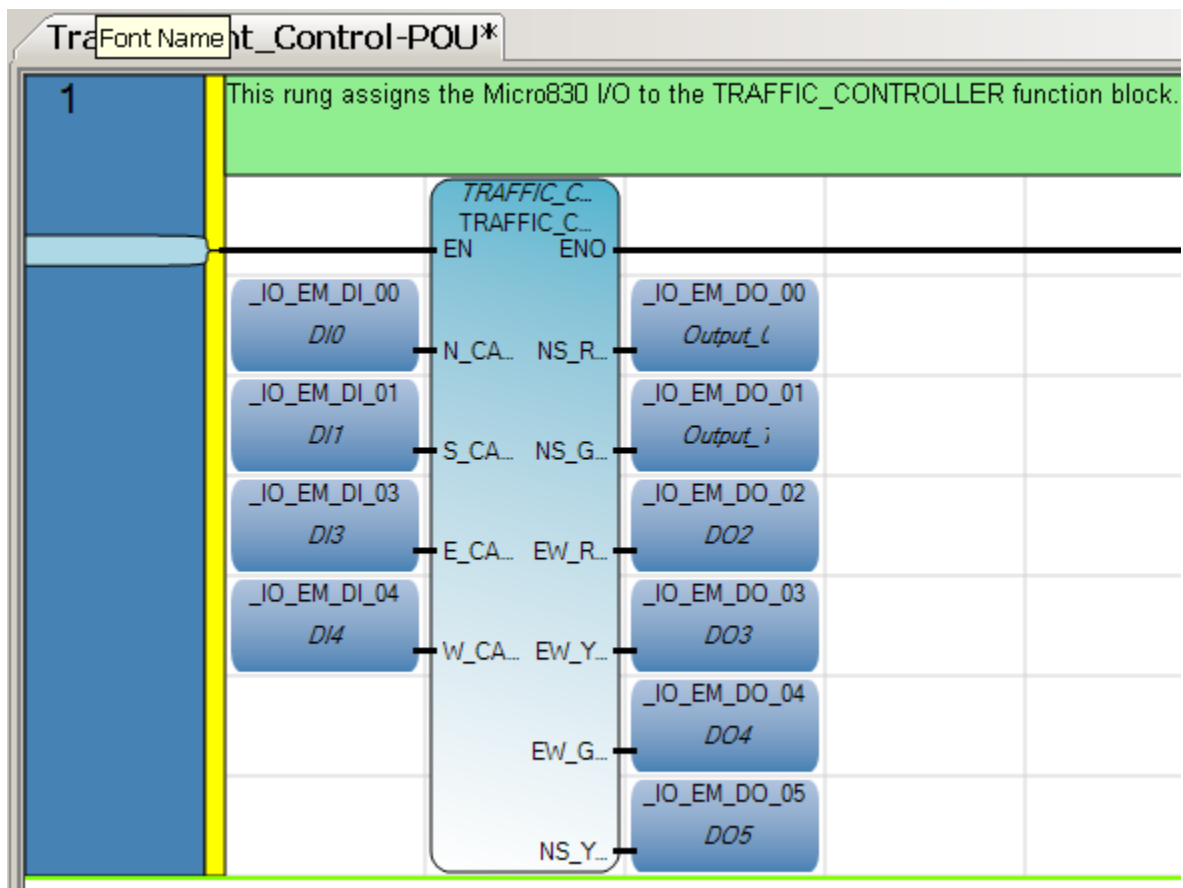
15. Assign the remaining three input variable blocks as follows: **_IO_EM_DI_01** to **S_Car...**, **_IO_EM_DI_03** to **E_Car...**, and **_IO_EM_DI_04** to **W_Car...** (note that we skipped using Input 2!):



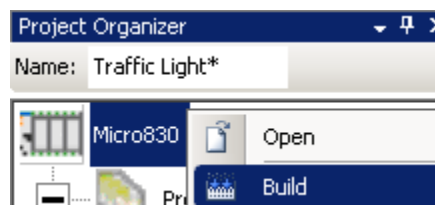
16. The first function block output that connects directly to the ladder rung is the function block output enabled (**ENO**) bit – it reflects the status of the input enable (**EN**) bit. The remaining six function block outputs are “real world” outputs that connect to the red, yellow and green traffic signal lights for each direction. These outputs get mapped to six Boolean output variables local to the function block: **NS_GREEN_LIGHTS**, **NS_YELLOW_LIGHTS**, **NS_RED_LIGHTS**, **EW_GREEN_LIGHTS**, **EW_YELLOW_LIGHTS**, and **EW_RED_LIGHTS**. Assign the first six Micro830 digital outputs to the output variable blocks starting with **_IO_EM_DO_00** to **NS_R...** and ending with **_IO_EM_DO_05** to **EW_G...**:



17. The rung is now complete except for a description of what the rung does. Double click on the green area just above the rung and type in “**This rung assigns the Micro830 I/O to the TRAFFIC_CONTROLLER function block.**”:



18. Finally, build and save the one-rung program. Right click on the Micro830 icon in **Project Organizer** and select **Build**:



33. You should get verification in the **Output** window at the bottom center of the screen that the build succeeded:

```

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

```

Click the Save icon  to save your work.