

# **Connected Components Workbench Task-based Quick Start**

## **QS# GS03 – Creating a User Defined Function Block (UDFB)**

## **Hardware & Software Versions Used to Develop This Quick Start**

- CCW Release 1, Build 51
- 2080-LC30-16QWB v1.03

## **Prerequisite**

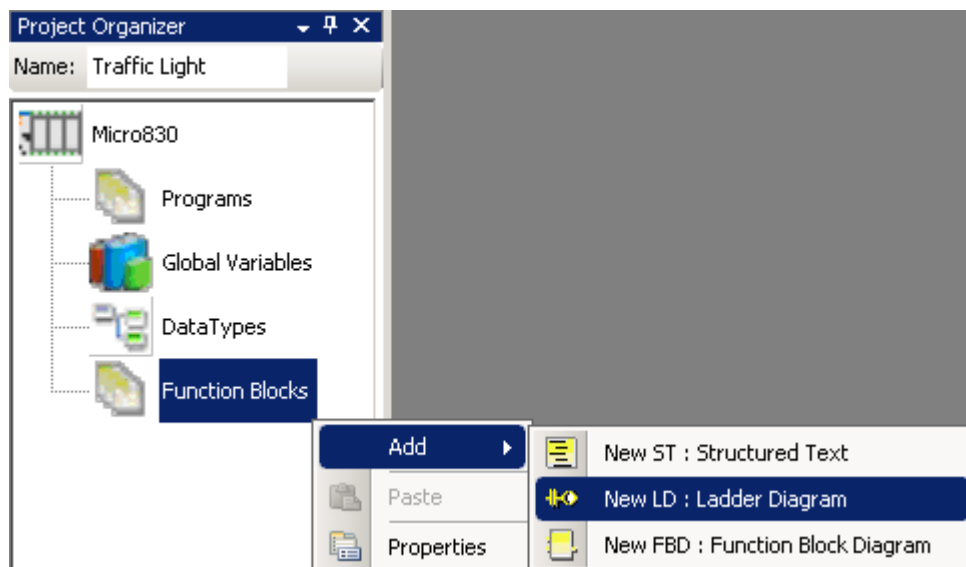
- QS# GS01 – Creating a New CCW Project

---

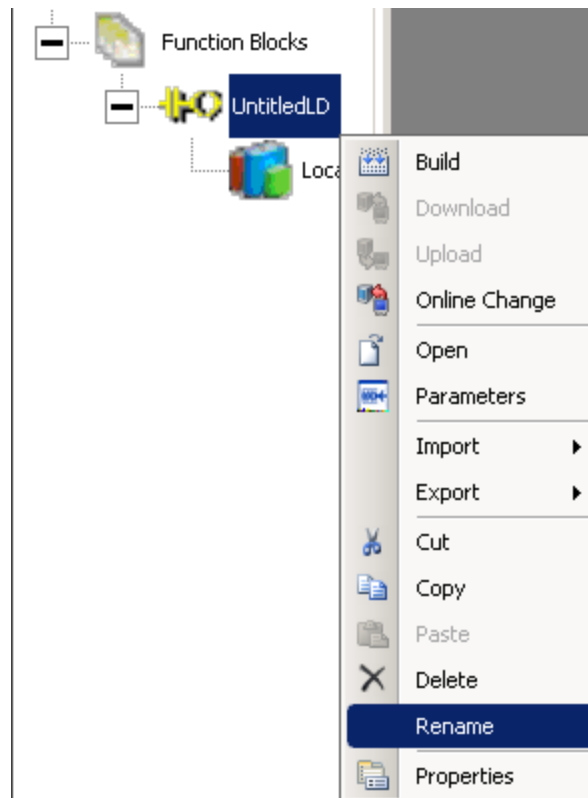
## Creating a User Defined Function Block (UDFB)

This quick start will show you how to create a User Defined Function Block for controlling a Traffic Light.

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



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



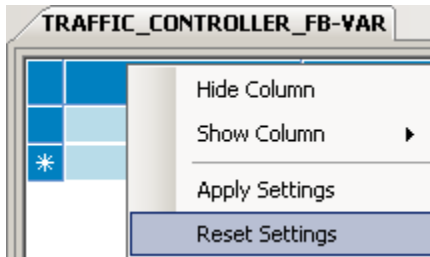
3. Type **TRAFFIC\_CONTROLLER\_FB** (the name given to the Ladder Diagram file will be the name of the UDFB) and Enter:



4. The general Traffic Controller algorithm for the function block to be implemented in the ladder diagram program is as follows:  
If a car is waiting at a red light for at least five seconds, change the green lights in the other direction to yellow for two seconds and then to red, while changing the red lights in the waiting car's direction to green.
5. The most important thing to define up front when creating a UDFB is what inputs are required, and what outputs will be produced, by this function block. These inputs and outputs are defined in the function block's **Local Variables**. Therefore, under **TRAFFIC\_CONTROLLER\_FB**, double click on **Local Variables**:



- Right click on the top row and select **Reset Settings** so all of the necessary columns are displayed:

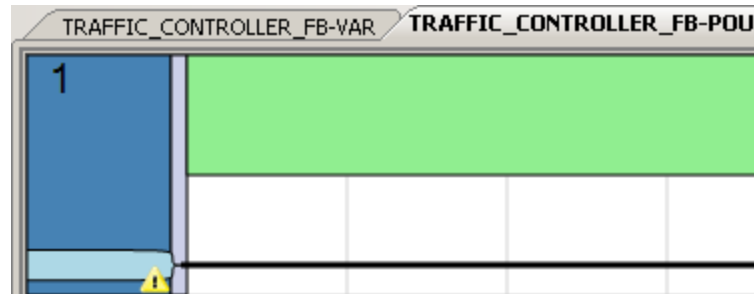


- For this UDFB, we need four Boolean inputs (for a car sensor in each of the four positions in the intersection) and six Boolean outputs (for red, yellow and green lights in each of the two directions). The inputs are entered in with **Direction VarInput** and the outputs are entered in with **Direction VarOutput**. Enter in the variables with the **Names**, **Data Types** and **Directions** as shown:

Name	Data Type	Dimension	Alias	Comment	Initial Value	Direction
N_CAR_SENSOR	BOOL					VarInput
S_CAR_SENSOR	BOOL					VarInput
E_CAR_SENSOR	BOOL					VarInput
W_CAR_SENSOR	BOOL					VarInput
NS_RED_LIGHTS	BOOL					VarOutput
NS_YELLOW_LIGHTS	BOOL					VarOutput
NS_GREEN_LIGHTS	BOOL					VarOutput
EW_RED_LIGHTS	BOOL					VarOutput
EW_YELLOW_LIGHTS	BOOL					VarOutput
EW_GREEN_LIGHTS	BOOL					VarOutput

- Double click on **TRAFFIC\_CONTROLLER\_FB** to begin editing the ladder logic program.





9. We want the first rung to work as follows:

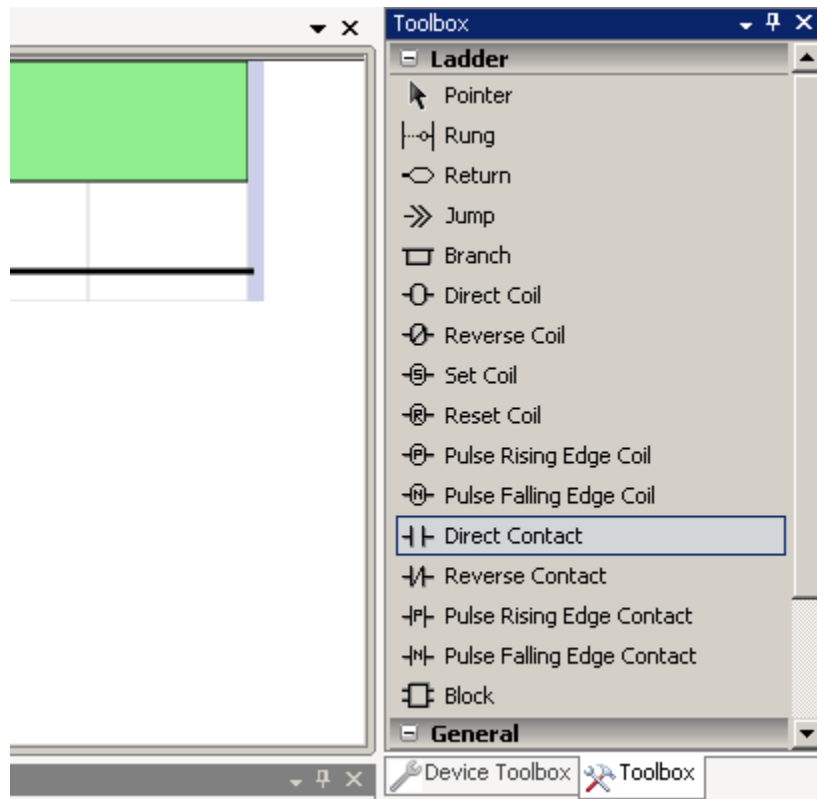
If:

the North/South Red Lights and East/West Green Lights are on, and  
 a car trips either the North Sensor or the South Sensor for at least five seconds,

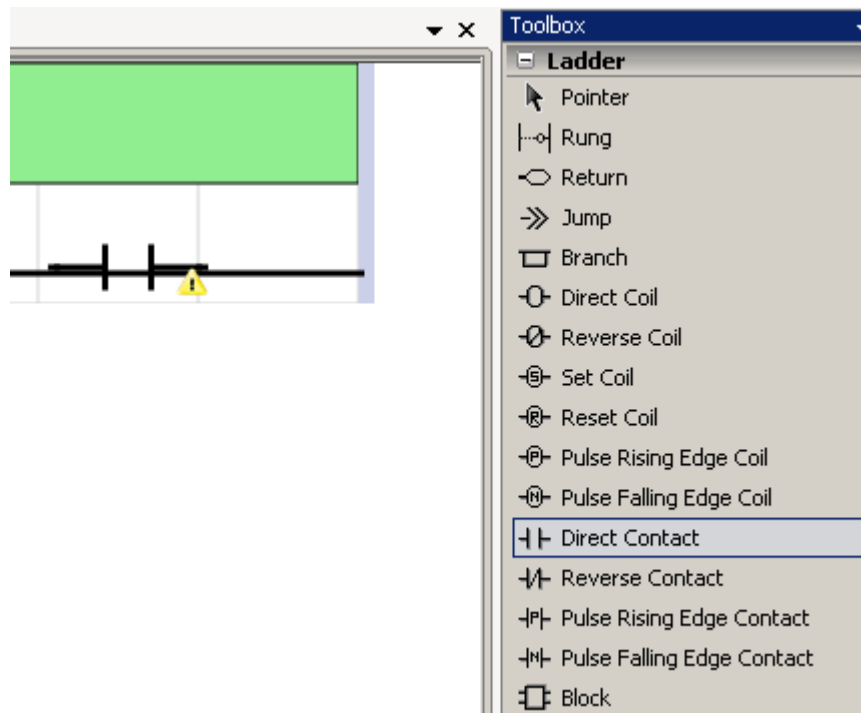
Then:

change the East/West Lights from Green to Yellow.

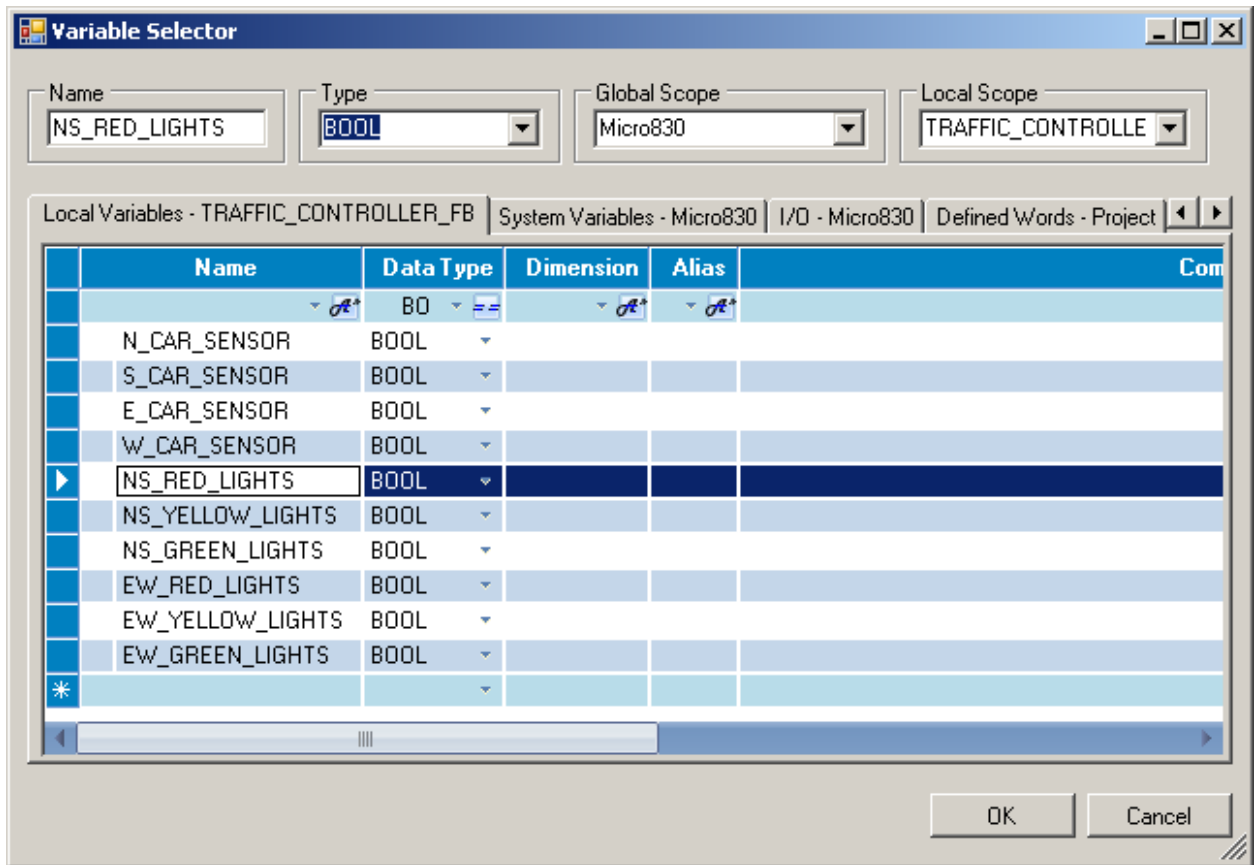
10. Click on the **Toolbox** tab in the lower right-hand corner and click on the **+** in front of **Ladder** to list the available ladder instructions:



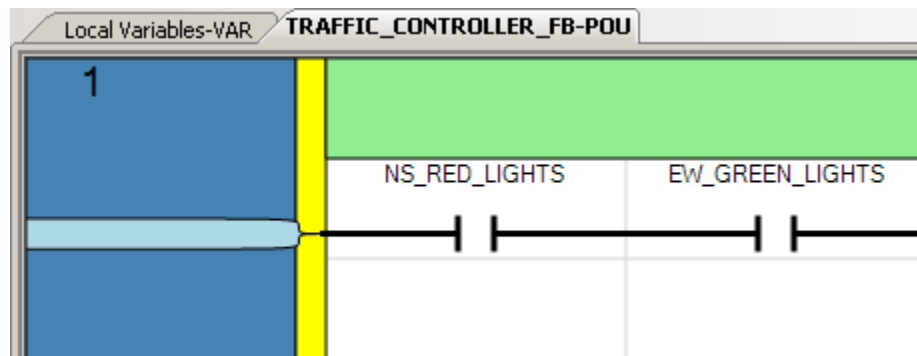
11. To implement the first bullet, we need two **Direct Contacts** in series (since the logic is North/South Red Lights *AND* East/West Green Lights). Click and drag a **Direct Contact** instruction from the **Toolbox** to the rung and release:



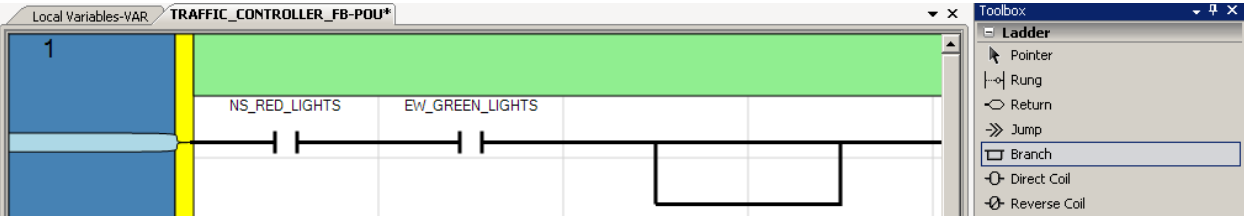
12. When you release the mouse button, the **Variable Selector** screen appears. Select **NS\_RED\_LIGHTS** and click **OK**:



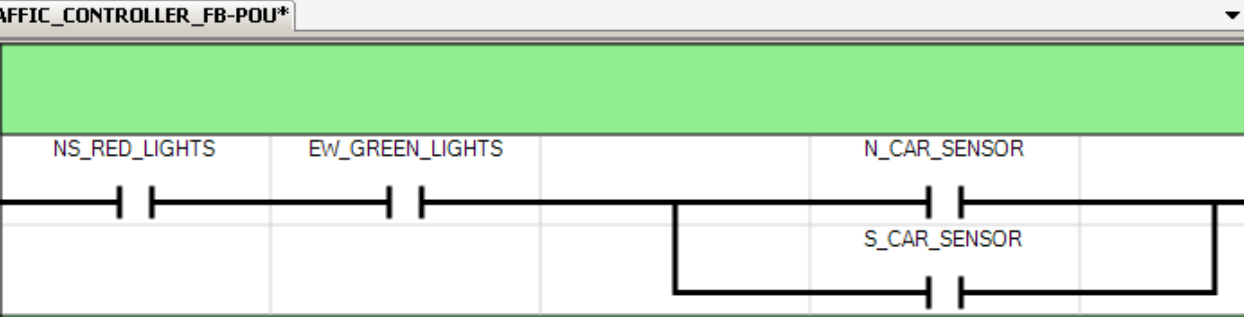
13. Similarly, add a second **Direct Contact** and assign **EW\_GREEN\_LIGHTS** to this contact. So far, your rung should look like this:



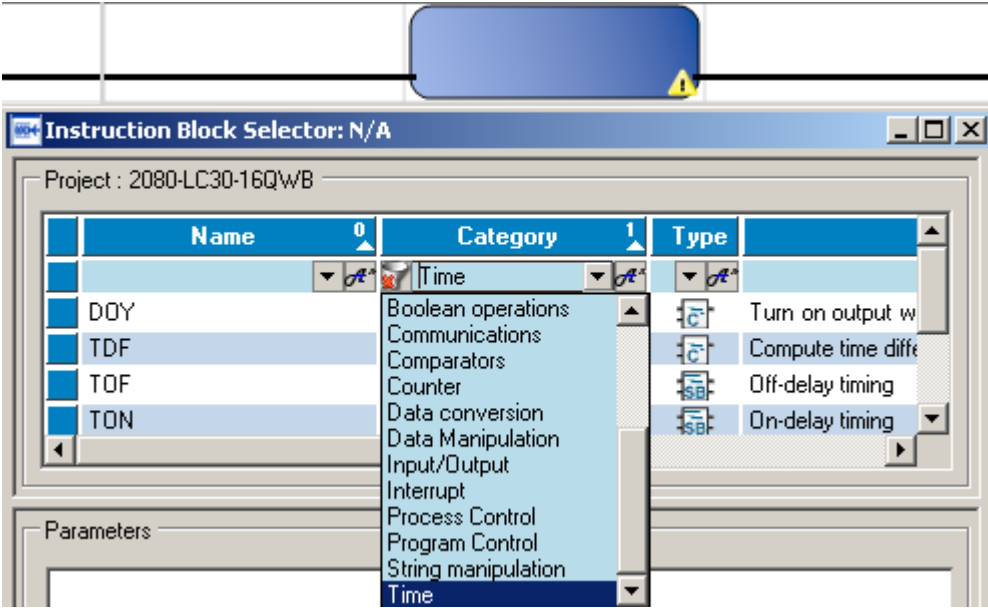
14. To implement the second bullet, we need two **Direct Contacts** in parallel (since it's North Car Sensor *OR* South Car Sensor). First, click and drag a **Branch** from the **Toolbox** to the end of the rung and release:



15. Click and drag a **Direct Contact** instruction from the **Toolbox** to the upper branch and release. Assign variable **N\_CAR\_SENSOR** to this contact. Then, click and drag a **Direct Contact** instruction from the **Toolbox** to the lower branch and release. Assign variable **S\_CAR\_SENSOR** to this contact. Now your rung should look like this:



16. Next we need a five second time delay, so click and drag a **Block** instruction from the **Toolbox** to the right of the branch and release. The **Instruction Block Selector** screen appears. Under the **Category** column, select **Time** to list all the Time-based instructions:



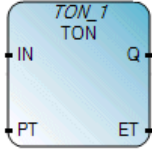
Select TON for On-delay timing and click OK.

17. Click on the **TON** block and hit the **F1** key to bring up context-sensitive help in order to get an explanation of the block inputs and outputs:

**TON**

**Description:**

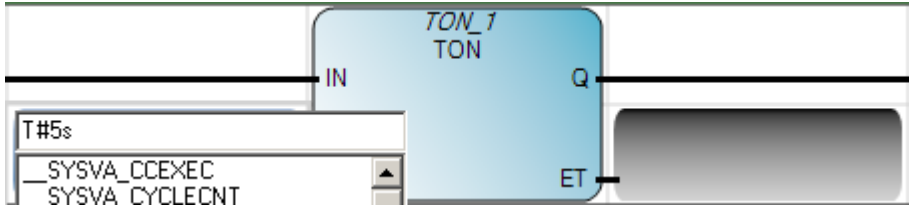
Increase an internal timer up to a given value.



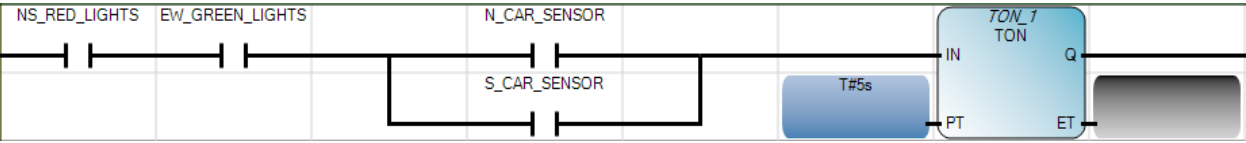
**Arguments:**

Parameter	Parameter Type	Data Type	Description
IN	Input	BOOL	If rising edge, starts increasing internal timer. If falling edge, stops and resets internal timer.
PT	Input	TIME	Maximum programmed time.
Q	Output	BOOL	If TRUE, programmed time is elapsed.
ET	Output	TIME	Current elapsed time. Possible values range from 0ms to 1193h2m47s294ms.

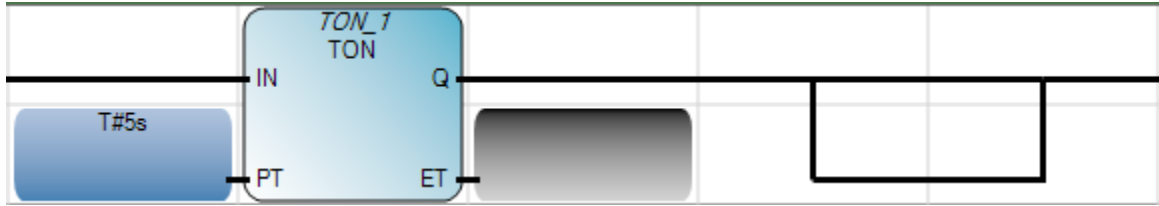
18. Click on the top of the **PT** input block and enter in a programmed/preset time of “T#5s” (for a time format of 5 seconds) as shown below and enter:



Now your rung should look like this:



19. To implement the last bullet, we need two **Coils** in parallel (since we need to turn off the East/West green lights *AND* turn on the East/West yellow lights). First, click and drag a **Branch** from the **Toolbox** to the end of the rung and release:



20. Click and drag a **Reset Coil** instruction from the **Toolbox** to the upper branch and release. Assign variable **EW\_GREEN\_LIGHTS** to this coil (this will turn off the East/West green lights). Then, click and drag a **Set Coil** instruction from the **Toolbox** to the lower branch and release. Assign variable **EW\_YELLOW\_LIGHTS** to this coil (this will turn on the East/West yellow lights). Now your completed rung should look like this:



21. Now document the rung by double clicking in the green block above the rung and entering in "If the North/South red lights and East/West green lights are on and a car trips either the North sensor or the South sensor for at least five seconds, then change the East/West lights from green to yellow."

22. We want the second rung to work as follows:

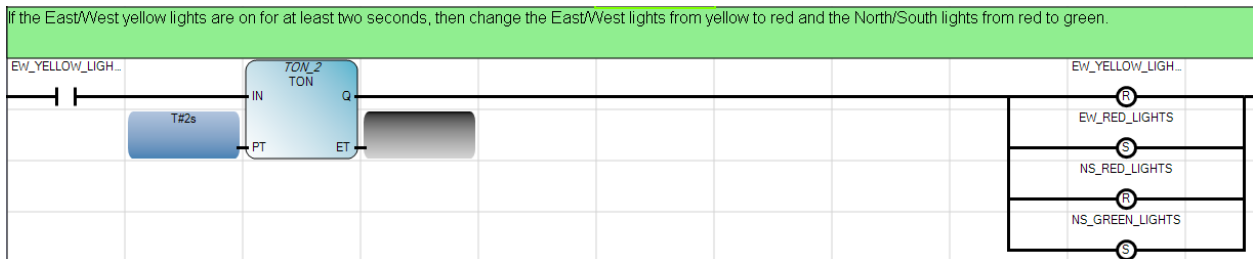
If:

the East/West Yellow Lights are on for at least two seconds,

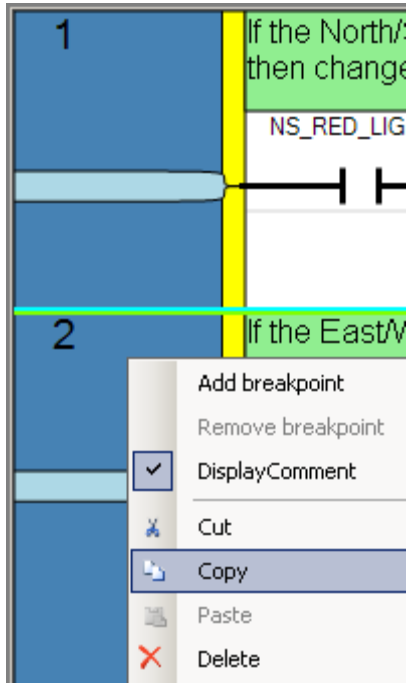
Then:

change the East/West Lights from Yellow to Red and the North/South Lights from Red to Green.

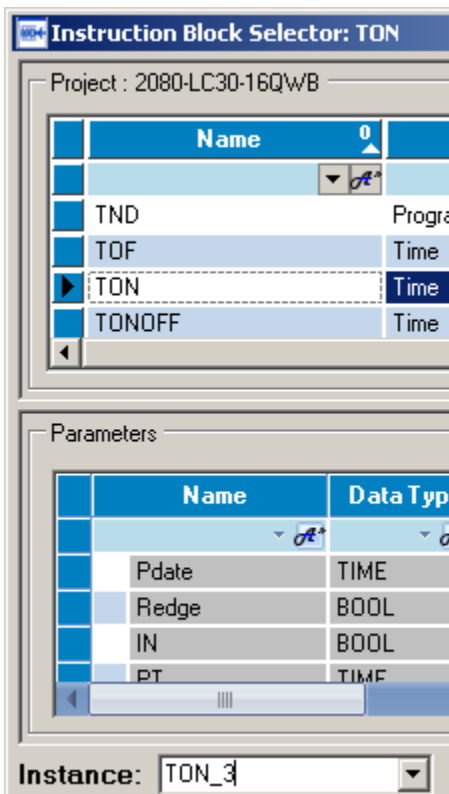
23. Enter in the second rung to look like the following:



24. The next two rungs have the same form as the first two rungs, so we will take advantage of cut and paste, then just edit the variable assignments for each instruction. Select the two rungs by clicking in the dark blue region to the left of rung 1, then hold the **Shift** key down and click in the dark blue region to the left of rung 2. Right click and select **Copy**:

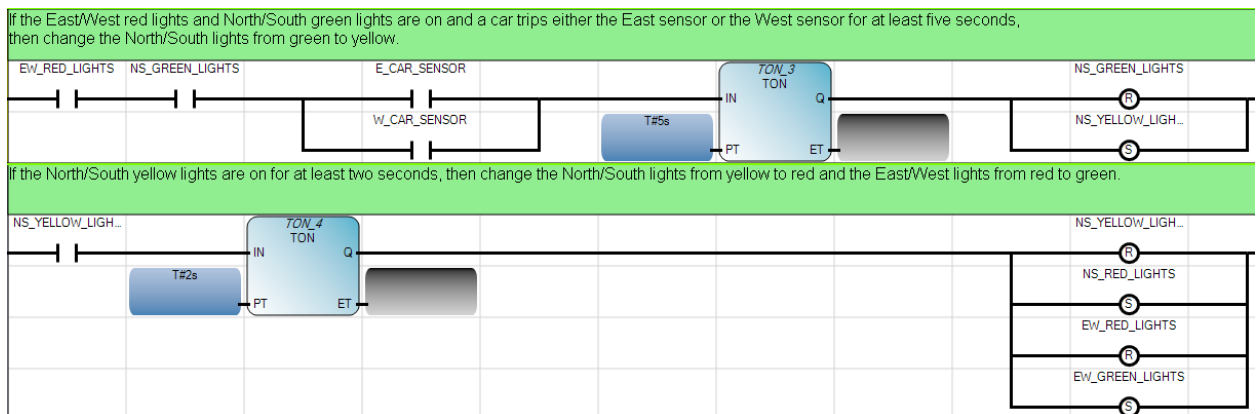


25. Now select just rung 1, then right click and **Paste**. The two rungs are inserted prior to the original two rungs. Double click on the **TON** in rung 3 and within the **Instruction Block Selector** screen, edit the **Instance** from **TON\_1** to **TON\_3** and click **OK**:

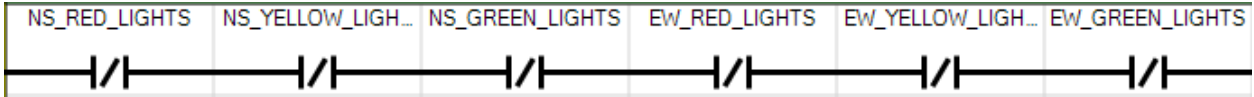


Similarly, change the **Instance** of the **TON** in rung 4 from **TON\_2** to **TON\_4**.

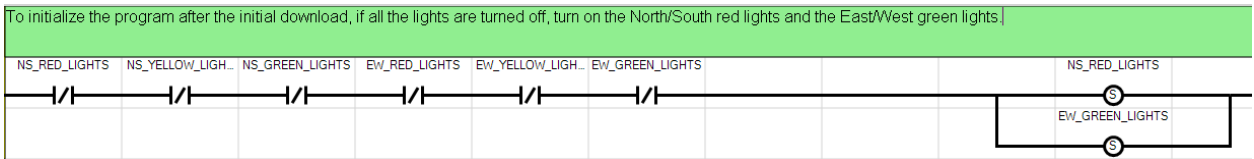
26. In rungs 3 and 4, change each **EW** variable to **NS** and each **NS** variable to **EW**, so that the rungs appear as follows (don't forget to modify the rung comments to match!):



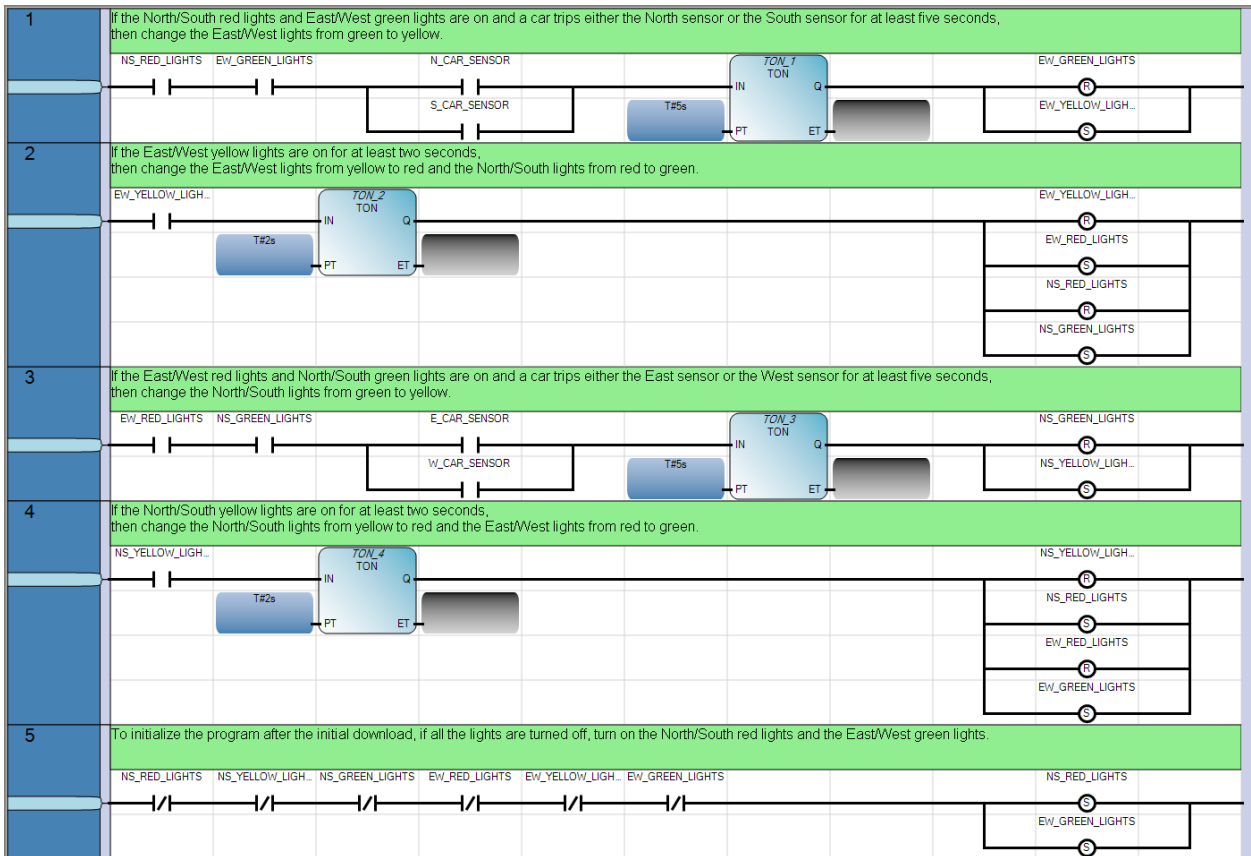
27. We need to add one more rung to handle “initial conditions”. When the program is first downloaded to the controller and run, none of the lights are initially turned on. This last rung will check for this condition (all lights off) and turn on the North/South red lights and the East/West green lights. Click and drag a **Rung** from the **Toolbox** to the white space below rung 4 and release. Click and drag six **Reverse Contacts** from the **Toolbox** onto the new rung and assign one ...**LIGHTS** variable to each contact:



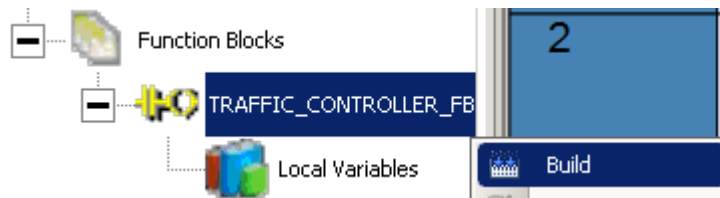
28. Next, add two **Set Coils** in parallel to turn on **NS\_RED\_LIGHTS** and **EW\_GREEN\_LIGHTS**. Complete rung 5 by documenting its operation:



29. Finally, build and save the five-rung program.



Right click on **TRAFFIC\_CONTROLLER\_FB** in **Project Organizer** and select **Build**:



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

```

Output
Show output from: Build
Compiling for 2080-LC30-16QWB
TRAFFIC_CONTROLLER_FB
    0 error(s), 0 warning(s)
Compiling for SIMULATOR
TRAFFIC_CONTROLLER_FB
    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.