

ILC LightSync Expanded Conditional Input Software Manual

Rev B

Table of Contents

Section 1 Program Description

- 1.0 Overview
- 1.1 Suggested Minimum System Requirements
- 1.2 Starting the Program
- 1.3 Navigating the Program

Section 2 Connecting to the Lighting System

- 2.0 Overview
- 2.1 Interfacing the Software to the Hardware

Section 3 Programming

- 3.0 Overview
- 3.1 Creating “AND” Conditional Logic
- 3.2 Configuring an “OR” Conditional Logic
- 3.3 Testing Configured Logic
- 3.4 Text Notes

Section 4 File Management

- 4.0 Overview
- 4.1 Compile Program
- 4.2 Download Programming
- 4.3 Uploading From Module

Section 1 Program Description

1.0 Overview

ILC LightSync Expanded Conditional Input Software is an ILC proprietary software package designed for configuring the “AND” and “OR” logic for the LightSync Conditional Input Module (LS-CIM) and the LightSync Conditional Occupancy Sensor 8 Input Module (LS-COS8IM). The software provides a 16 by 16 matrix for building the logic conditionals for the 8 inputs on the module. Once designed, the data is compiled and downloaded to the Conditional Input Module.

1.1 Suggested Minimum System Requirements

The following are the minimum system requirements required to run the LightLEEDer Expanded Conditional Input Software. Running this software with less than the minimum hardware requirements or operating systems other than listed may provide unexpected results.

- Personal Computer or Laptop
- 2.4 GHz core 2 duo processor
- Windows 7, or 10
- 32- or 64-bit OS
- USB 1.0 or 2.0 port
- 2Gb Ram
- Display Resolution 1024 x 768
- ILC LightLEEDer Expanded Conditional Input software installed

Note:

The LightLEEDer Expanded Conditional Input Software cannot be open at the same time with any other ILC software. Having both software packages open will cause changes to the other when saving.

1.2 Starting the Program

Double click the ILC LL ExpCnd.exe file provided to start the program. See figure 1 for opening screen.

Note: The ILC LightLEEDer configuration software needs to be installed prior to running the program.

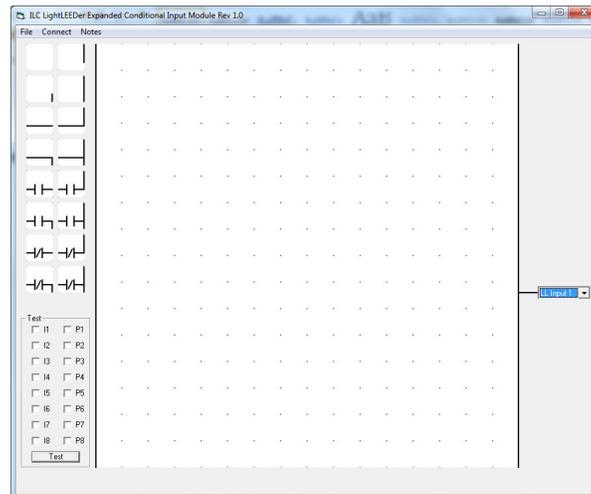


Figure 1 Main Opening Screen

1.3 Navigating the Program

Title bar:

- **File:** This pull-down menu allows you to create a new file, open an existing file, save a file, save-as for renaming a current file, and Exit
- **Connect:** This pull-down menu has options to compile the data, download, upload, and selection of the communications port.
- **Notes:** This opens a box for notes on the conditions programmed.

Program Body:

- **Logic Icons:** On the left side of the screen there are drag-and-drop gate icons. These are used to represent inputs/states gates and connections to building the logic.
- **Logic Test:** In the lower left of the screen you can set the input or pilot states to test the logic for the switch input.
- **Logic Matrix Field:** The center field is where the logic matrix is built. There are 16 x 16 cells that the icons can be dragged into.
- **Input:** This is a pull-down menu located on the right side of the screen. This selects the input to be programmed.

Section 2 Connecting to the Lighting System

2.0 Overview

The LightSync Expanded Conditional Input software requires a LightSync Conditional Module Programming Pack (LSCM-PP) for interfacing and programming this conditional input module. The module communicates through the modules CAT5 data line port via the PC's USB port and a CAT5 cable as shown in figure 2.

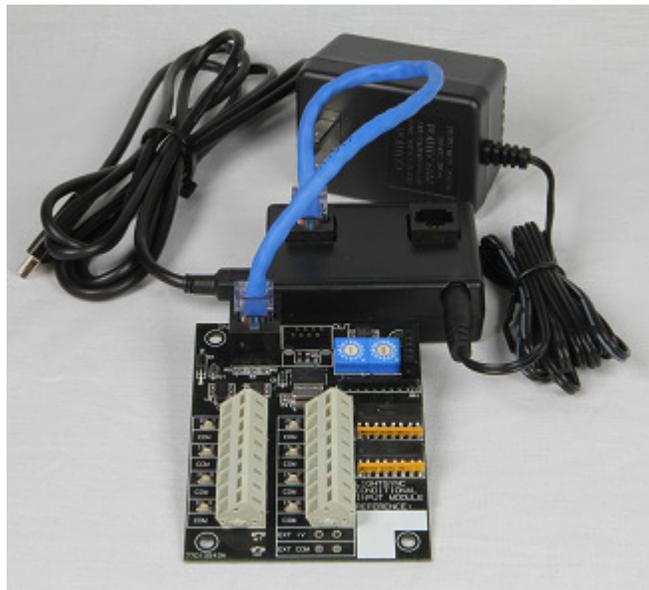


Figure 2 LSCM-PP Connected to a Panel Mount Module

2.1 Interfacing the Software to the Hardware

1. For panel mount modules, turn the power OFF to the controller with the power switch located under the lower right hand side of the keypad. For remote modules remove all data line connections from the IN and OUT ports.
2. Set the rotary address switches on the device to address 00.
3. Connect a CAT5 cable from the IN port of the conditional input module to the OUT port on the LSCM-PP.
4. Plug the provided USB cable from the LSCM-PP to a PC with the software installed.
5. Connect the 12VDC wall transformer provided to the LSCM-PP.
6. Examine the computers Device Manager and take note of what com port is being used for the Silicone Labs device that will show up under "Ports".

Section 3 Programming

3.0 Overview

This section describes how to configure the conditionals in this module. Each of the 8 inputs can be configured for the conditionals desired. The examples below in section 3.1-3.2 show the basic AND and OR logic configurations. Each input can use multiple of logic circuits so the input responds the way it is desired. These logic inputs can be used in conjunction with input and output conditionals in the lighting panel. The software provides a 16 x 16 matrix for building the desired logic conditionals.



Figure 3 Logic Gate for a Normally Open Switch or Pilot status ON

Figure 3 shows the symbol that represents a currently OPEN switch or a pilot currently OFF. This switch needs to be CLOSED or the pilot needs to be ON to make the conditional true, once true, it will send a closure to the panel.



Figure 4 Logic Gate for Normally Closed Switch or Pilot status OFF

Figure 4 shows the symbol that represents a currently CLOSED switch or a pilot status currently ON. This switch needs to be OPEN or the pilot needs to be OFF to make the conditional true, once true, it will send a closure to the panel.

3.1 Creating “AND” Conditional Logic

This conditional logic contains 2 or more variables in series that need to be true to activate the switch closure at the panel. From the icons on the left side of the screen, drag-and drop the selected items to build the logic from the left to the right of the screen. Note: You must connect from the left side to the right side to complete the logic circuit.

As seen in figure 5 an “AND” logic was created for input 1 and pilot 3. For this input to send the signal to the panel, input 2 must be CLOSED and pilot 3 must be ON to make the logic true. If all of these are true, the switch signal will be sent to the panel.

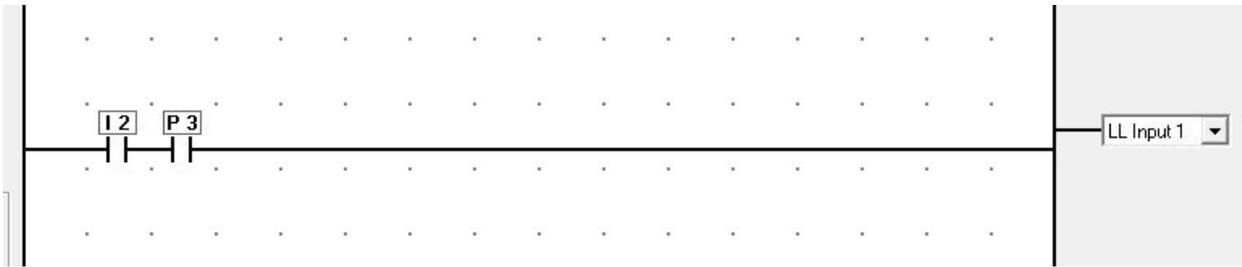


Figure 5 Simple “AND” Logic Conditional

3.2 Configuring “OR” Conditional Logic

This conditional logic contains 2 or more variables in parallel that only needs one to be true to activate the switch closure at the panel. From the icons on the left side of the screen, drag-and drop the selected items to build the logic from the left to the right of the screen. Note: You must connect from the left side to the right side to complete the circuit.

As seen in figure 6 an “OR” logic was created for input 1, input 2, and input 3. For this input to send the signal to the panel, input 1 could be OPEN, or either input 2 or 3 CLOSED. If any of these are true, the switch signal will be sent to the panel.

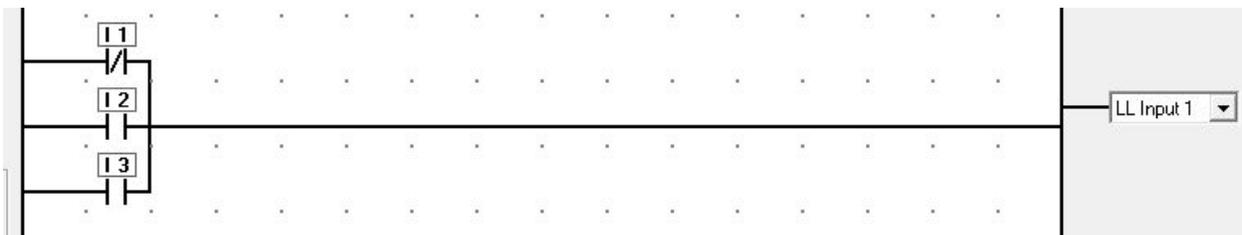


Figure 6 Simple “OR” Logic Conditional



Figure 7 Multiple Logic Conditional Example

3.3 Testing Configured Logic

The software is provided with the means to test the logic for each conditional input to verify the operation desired. The test allows you to simulate activation of a switch and pilot to verify the action.

1. From the right side of the screen, select the input to be tested.
2. In the left side of the screen select the inputs and pilots to be active for this test as shown in figure 8.



Figure 8 Activate Input/Pilot

3. Press the "TEST" button to test the conditional logic.
4. A window will open with the results as shown in figure 9 showing the logic is true and the switch closure is sent to the panel.
5. Press "OK" to close the window.

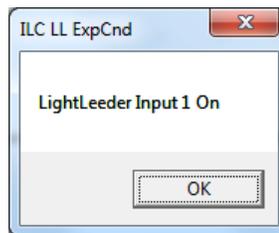


Figure 9 Test Result

3.4 Text Notes

Notes can be created for the programming and is saved in the programming file. Note: This is not downloaded into the module and only resides on the computer.

Section 4 File Management

4.0 Overview

Once the conditional programming has been designed and tested it is time to compile, save, and download it to the module.

4.1 Compile Programming

1. Select “Connect” pull-down menu as shown in figure 11, and then select the proper com port discovered in the procedure in section 2.1.
2. From the same “Connect” pull-down menu select “Compile”, you will see the status at the bottom of the screen as shown in figure 12 and a pop-up window will indicate completion.

4.2 Saving Programming

1. From the title bar as shown in figure 10 select “File”, and then “Save As”. Name the file with a unique name and save. Once a file has been saved to a file, the “Save” option can be used. The program will be saved as a .CDF file.

4.3 Downloading Logic Programming

1. From the “Connect” pull-down menu as shown in figure 11 select “Download” to send the programming to the module. Note: Be sure to follow the procedure in section 2.1 before attempting a download. Figure 13 shows the download progress window located in the upper right hand corner during the process.
2. Once the download is complete, a download complete window will appear. The module needs to be power cycled to operate properly.

4.4 Uploading Logic Programming

The programming, if needed, can be uploaded from the module. The procedure is very similar to the download procedure.

1. From the “Connect” pull-down menu as shown in figure 11 select “Upload” to pull the programming up to the computer from the module. Note: Be sure to follow the procedure in section 2.1 before attempting an upload. Figure 12 shows the upload progress window located in the upper right hand corner during the process.
2. You will receive the upload message shown in figure 14.
3. Once the upload is complete, an upload complete window will appear. The module needs to be power cycled to operate properly.

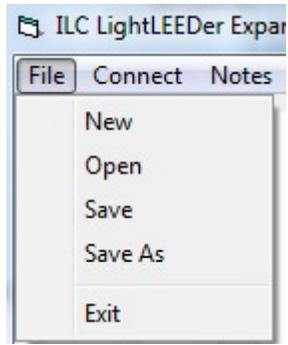


Figure 10 File Pull-Down Menu

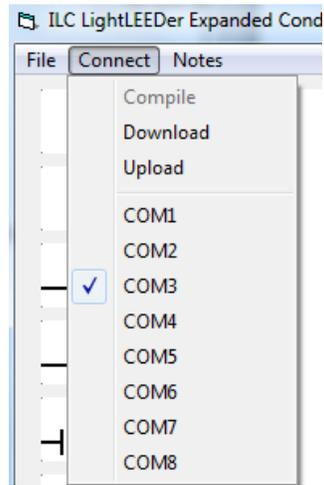


Figure 11 Connect Pull-Down Menu

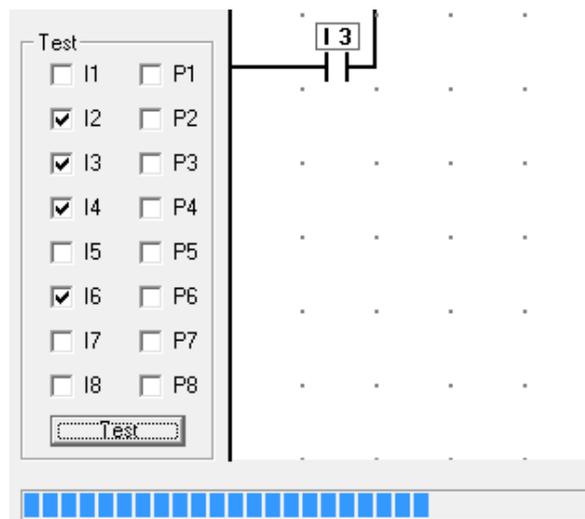


Figure 12 Compile Progress Indicator

Block 272 of 2304

Figure 13 Download/Upload Progress

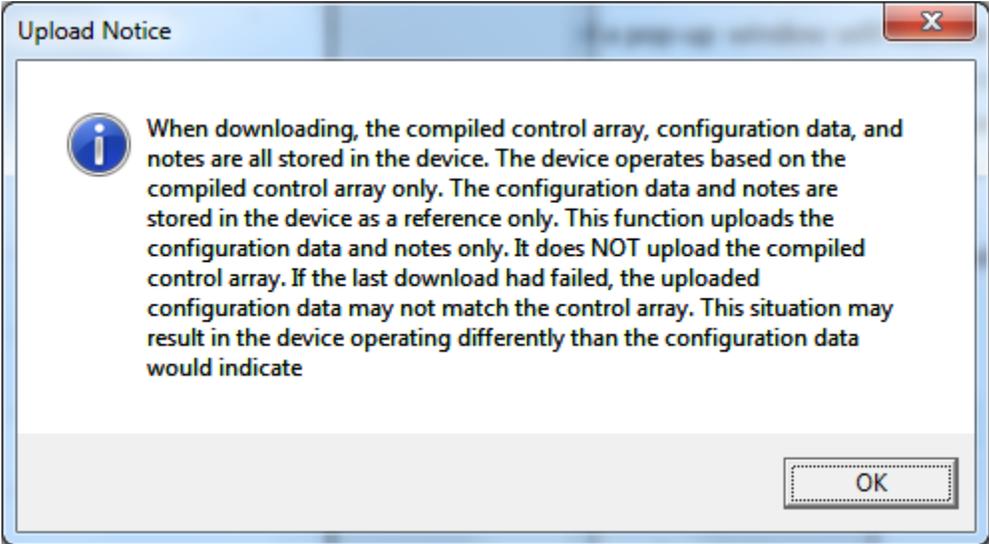


Figure 14 Upload Notice