c. Select the correct Com Port number where the programming cable is connected to the computer then press the “Link” button.

![Link Com Port](image)

5. Write simple program

a. Write a simple one rung program by clicking on the leftmost cell at line 001 of the programming grid, then click on the “M” contact icon on the ladder toolbar, as shown below. Select M1 and press the OK button. See Chapter 4: Ladder Programming instructions for complete instruction set definitions.

![Programming Grid](image)

Note: If the ladder toolbar is not visible at the bottom of the screen, select View>Ladder Toolbar from the menu to enable.
b. Use the “A” key on your keyboard (or the “A” icon from the ladder toolbar) to draw the horizontal circuit line from the M contact to the right most cell, as shown below.

c. Select the “Q” coil icon from the ladder toolbar and drop it on the right most cell. Select Q1 from the dialog and press OK as shown below. See Chapter 4: Ladder Programming instructions for complete instruction set definitions.
d. Test the simple program. From the Operation menu, select the Write function and write the program to the connected PLR as shown below.

e. Select the RUN icon from the toolbar, and select “No” when the pop-up message asks “Do you want to read program from module?”, as shown below.
f. From the Input Status dialog, click on M1 to activate the contact M1 which will turn ON the Output Q1, as shown below. The highlighted circuit will show active and the first Output (Q1) on the connected PLR will be ON. See Chapter 3: Programming Tools for more detailed software information.
Left Blank
Chapter 3: Program Tools

New Ladder Program
Select File-->New-->New LAD to enter the development environment for a new Ladder program.

New FBD Program
Select File-->New-->New FBD to enter the development environment for a new FBD (Function Block Diagram) program.

Open Existing File
Select File-->Open to choose the type of file to open (Ladder or FBD), and choose the desired program file, and then click Open.

Ladder Logic Programming Environment
The Ladder Logic Programming Environment includes all the functions for programming and testing the SG2 PLR using the Ladder Logic programming language. To begin a new program select File-->New--> and select the desired model of SG2, and the number of connected expansion units if applicable, as shown below.
Menus, Icons and Status Displays

The Ladder programming environment includes the following Menus, Icons and Status Displays

1. Menu bar – Five menu selections for program development and retrieval, editing, communication to connected controllers, configuration of special functions and viewing preference selections.

2. Main Toolbar – (From Left to Right)
   Icons for a New program, opening a program, and printing
   Icons for Keypad, Ladder view, HMI/Text editing and Symbol (comments) editing.
   Icons for Monitor, Simulator, Controller Mode changes (Run, Stop, and Quit), and Read/Write programs to/from the PLR.

3. Usage List – List for all memory types and addresses used with the current open program. Used addresses are designated by a “*” symbol below each address.

4. Amount of free programming memory available.

5. Current Mode – operation mode of the controller, or simulator, from the connected PC.


7. Status Bar – Status of current open project and connected PLR.
Programming

The SG2 Client software can be programmed by either drag-and-drop of instructions or by using keyboard entry commands. Below is an example of some common methods of entering programming instructions.

The “A” and “L” keys or icons are used to complete parallel and serial circuits. The rightmost column is for output coils.
Simulation Mode

The SG2 Client software includes a built-in simulator to test and debug programs easily without the need for downloading to a controller. To activate simulation mode, simply press the red RUN icon. The program below is shown in simulation mode, identifying the significant available features.

Establish Communication

The following is the simple procedure for establishing communication between the connected PC and the SG2 PLR.

a. Select “Operation/Link Com Port…” as shown below.
b. Select the correct Com Port number where the programming cable is connected to the computer then press the “Link” button.

![Link Com Port](image)

```
Link Com Port

- COM1 PORT
- COM2 PORT
- COM3 PORT
- COM4 PORT
- COM5 PORT
- COM6 PORT
- COM7 PORT
- COM8 PORT
```

```
[Link] [Unlink] [Close]
```

c. The SG2 Client software will then begin to detect the connected PLR to complete it’s connection as shown below.

![Please Wait a Moment!](image)

```
Please Wait a Moment!
Now Detect SG2 ID = 1
```

### Writing Program to PLR

From the Operation menu, select the Write function and write the program to the connected PLR as shown below.

![LAD Version: 1.4](image)

```
LAD Version: 1.4

File  Edit  Operation  View  Help

Monitor
Simulator

Coil/Symbol

Run ! Ctrl+R
Stop ! Ctrl+T
Quit Ctrl+Q

Read

Write

X: 1234

RTC Set...
Password...
Language...
Module System Set...
Link Com Port...
```

---

SG2 PLR User Manual  www.electrodepot.com
Operation menu

The Operation menu, includes several system configuration functions for both online and offline setup. The following explains the details of each function.

Monitor – Online function for runtime monitor and editing when connected to a controller

Simulator – Offline function for testing and debugging a program.

Run-Stop-Quit – Mode change selections for both runtime editing and simulation mode.

Read-Write – Reading and writing programs to and from a connected PLR.

RTC Set – Online function for setup of the Real-time clock/calendar (see dialog below)

Password – Set a password for accessing the current program after upload to the PLR

Language – Change software language

Module System Set – Dialog for changing important system setup functions including Module ID, Remote I/O preferences, Expansion I/O settings, and Retentive memory preferences (Keeping) for (C) Counters, (M) Auxiliary Coils, and the LCD Backlight.

Online Monitoring/Editing

The SG2 Client software allows for online monitoring of the currently running program during runtime. Additional online functions include, I/O forcing, and Mode changes (Run/Stop/Quit).

Note: The SG2 Client software does not support runtime logic editing changes. All logic edits to contacts, coils, timers/counters, and circuit connecting lines must be written to the connected PLR while in Stop mode.
Program Documentation
The SG2 Client software includes the ability to document a program using Symbols and Line Comments. Symbols are used to label each I/O address up to a length of 12 characters. Line Comments are used to document sections of a program. Each Line Comment can have up to 4 lines with each line containing up to 50 characters in length. Below are examples of entering Symbols and Line Comments.

Symbol
The Symbol editing environment can be access through the menu using the Edit>Symbol… selection or using the symbol icon on the main toolbar shown below.

The Symbol editing environment allows for documenting all the contact and coil memory types, and selecting display modes as shown below.

Line Comments
Chapter 4: Relay Ladder Logic Programming

Common Memory Types

Inputs (I Memory Type)
The SG2 digital input points are designated I memory types. The number of digital I input points are 6, 8, or 12 depending on each SG2 model.

Outputs (Q Memory Type)
The SG2 digital output points are designated Q memory types. The number of digital Q output points are 4 or 8 depending on each SG2 model. In this example, output point Q1 will be turned on when input I1 activated.

Auxiliary Relays (M Memory Type)
Auxiliary relays are digital internal memory bits used to control a ladder logic program. The auxiliary relays are not physical inputs or outputs that can be wired to any external device; switches, sensors, relays, lamps, etc. Since auxiliary relays are internal bits within the CPU, they can be programmed as digital inputs (contacts) or digital outputs (coils). In the first rung of this example, auxiliary relay M1 is being used as an output coil and will energize when input I2 turns on. In the second rung auxiliary relay M1 is being used as an input and when energized, will turn on outputs Q2 and Q3.

Timers and Timer Status Bits (T Memory Type)
Timer status bits provide the relationship between the current value and the preset value of a selected timer. The timer status bit will be on when the current value is equal or greater than the preset value of a selected timer. In this example, when input I3 turns on, timer T1 will start. When the timer reaches the preset of 5 seconds timer status contact T1 turns on. When T1 turns on, output Q4 turns on. Turning off I3 will reset the timer.
Counters and Counter Status Bits (C Memory Type)

Counter status bits provide the relationship between the current value and the preset value of a selected counter. The counter status bit will be on when the current value is equal to or greater than the preset value of a selected counter. In this example, each time the input contact I4 transitions from off to on, the counter (C1) increments by one. When the counter reaches the preset of 2 counts, the counter status contact C1 turns on. When C1 turns on, output Q5 turns on. When M2 turns on counter C1 will reset. If M9 is turned on, the counter will change from a count-up counter to a count-down counter.

Common Memory Types

<table>
<thead>
<tr>
<th>Symbol</th>
<th>General output</th>
<th>SET output</th>
<th>RESET output</th>
<th>PULSE output</th>
<th>N.O. Contact</th>
<th>N.C. Contact</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input contact</td>
<td>( )</td>
<td>(↑)</td>
<td>(↓)</td>
<td>(P)</td>
<td></td>
<td></td>
<td>(N.O. / N.C.)</td>
</tr>
<tr>
<td>Output coil</td>
<td>Q Q Q Q</td>
<td>Q Q q</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8 (Q1-Q8 / q1-q8)</td>
</tr>
<tr>
<td>Auxiliary contact</td>
<td>M M M M</td>
<td>M M m</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15 (M1-MF / m1-mF)</td>
</tr>
<tr>
<td>Counter</td>
<td>C C c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15 (C1-CF / c1-cF)</td>
</tr>
</tbody>
</table>

Specialty Memory Types

<table>
<thead>
<tr>
<th>Symbol</th>
<th>General output</th>
<th>SET output</th>
<th>RESET output</th>
<th>PULSE output</th>
<th>N.O. Contact</th>
<th>N.C. Contact</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expansion input coil</td>
<td>( )</td>
<td>(↑)</td>
<td>(↓)</td>
<td>(P)</td>
<td></td>
<td></td>
<td>(N.O. / N.C.)</td>
</tr>
<tr>
<td>Expansion output coil</td>
<td>Y Y Y Y</td>
<td>X x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12 (X1-XC / x1-xC)</td>
</tr>
<tr>
<td>Differential (one shot)</td>
<td>Y Y Y Y</td>
<td>D (Positive)</td>
<td>d (Negative)</td>
<td></td>
<td></td>
<td></td>
<td>12 (Y1-YC / y1-yC)</td>
</tr>
<tr>
<td>RTC</td>
<td>R R r</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15 (R1-RF / r1-rF)</td>
</tr>
<tr>
<td>Analog comparator</td>
<td>G G g</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15 (G1-GF / g1-gF)</td>
</tr>
<tr>
<td>HMI</td>
<td>H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15 (H1-HF)</td>
</tr>
<tr>
<td>PWM</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 (P1)</td>
</tr>
<tr>
<td>DATA LINK</td>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8 (L1-LB)</td>
</tr>
</tbody>
</table>
Specialty Memory Types

Positive Input Differential Instruction (One-Shot)
An positive input differential instruction, or One-Shot, holds its status ON for one CPU scan when the preceding series contact transitions from OFF to ON. This transition from OFF to ON is called a Positive Input Differential.

Negative Input Differential Instruction (One-Shot)
A negative input differential instruction, or One-Shot, holds its status ON for one CPU scan when the preceding series contact transitions from ON to OFF. This transition from ON to OFF is called a Negative Input Differential.
Output Instructions

Set Output Instruction (Latch) (↑)
A set output instruction, or Latch, turns ON an output coil (Q) or an auxiliary contact (M) when the preceding input contact transitions from OFF to ON. Once the output is ON or set, it will remain ON until it is reset using the Reset output instruction. It is not necessary for the preceding input contact controlling the Set output instruction to remain ON.

Reset Output Instruction (Unlatch) (↓)
A reset output instruction, or Unlatch, turns OFF a previous set output coil (Q) or an auxiliary contact (M) when the preceding input contact transitions from OFF to ON. Once the output is OFF or reset, it will remain OFF until it is reset using another output instruction. It is not necessary for the preceding input contact controlling the Reset output instruction to remain ON.
Pulse Output Instruction (Flip-Flop) (P)

A pulse output instruction, or Flip-Flop, turns ON a coil (Q) or an auxiliary contact (M) when the preceding input contact transitions from OFF to ON. Once the output is ON, it will remain ON until the preceding input contact transitions from OFF to ON a second time. In the example below, When Pushbutton I3 is pressed and released Motor Q4 will turn ON and remain on. When Pushbutton I3 is pressed again, Motor Q4 will turn OFF and remain OFF. The pulse output instruction (P) will “flip-flop” its state from ON to OFF at each press of Pushbutton I3.
Counter Instructions

The SG2 PLR includes a total 15 separate counters that can be used throughout a program. Each counter has a choice of 8 operation modes, 6 for general purpose counting and 2 for high speed counting. Additionally, each counter has 6 parameters for proper configuration. The tables below describes each configuration parameter and lists each compatible memory type for configuring counters.

The figure to the right shows the relationship between the numbered block diagram for a Counter, the ladder diagram view, and the software Edit Contact/Coil dialog box.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Compatible Instructions</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Counting Mode (1-6)</td>
<td>Inputs</td>
<td>I1-Ic / i1-ic</td>
</tr>
<tr>
<td>2</td>
<td>Use (I1 ~ gF) to set counting up or counting down</td>
<td>Outputs</td>
<td>Q1-Q8 / q1-q8</td>
</tr>
<tr>
<td>3</td>
<td>OFF: counting up (0, 1, 2, 3, 4…..) ON: counting down ( ….3, 2, 1, 0)</td>
<td>Auxiliary coil</td>
<td>M1-MF / m1-mF</td>
</tr>
<tr>
<td>4</td>
<td>ON: the counter resets to zero and OFF</td>
<td>Expansion inputs</td>
<td>X1-XC / x1-xC</td>
</tr>
<tr>
<td>5</td>
<td>OFF: the counter continues to count</td>
<td>Expansion outputs</td>
<td>Y1-YC / y1-yC</td>
</tr>
<tr>
<td>6</td>
<td>Present Counting Value, range:0~999999</td>
<td>RTC</td>
<td>R1-RF / r1-rF</td>
</tr>
<tr>
<td></td>
<td>Target (Setting) Value, range:0~999999</td>
<td>Counter</td>
<td>C1-Cf / c1-cF</td>
</tr>
<tr>
<td></td>
<td>Code of the counter (C1 ~ CF total: 15 counters)</td>
<td>Timer</td>
<td>T1-TF / t1-tF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Analog comparator</td>
<td>G1-GF / g1-gF</td>
</tr>
</tbody>
</table>
Counter Mode 1 (Fixed Count, Non-Retentive)

Mode 1 Counter will count up to a fixed preset value and stop counting when the current count is equal to the preset value. Additionally, the current count value is non-retentive and will reset to zero on a loss of power to the PLR. In the example below, the counter will stop counting when it reaches the preset value of 20. Counter status bit C1 will be ON when the current value is 20.
Counter Mode 2 (Continuous Count, Non-Retentive)

Mode 2 Counter will count up to a fixed preset value and continue counting after the preset value. Additionally, the current count value is non-retentive and will reset to zero on a loss of power to the PLR. In the example below, the counter will continue counting after its preset value of 20. Counter status bit C1 will be ON when the current value is 20.
Counter Mode 3 (Fixed Count, Retentive)

Mode 3 Counter operation is similar to Mode 1 except its current count value is retentive. Mode 3 Counter will count up to a fixed preset value and stop counting at that value. Additionally, the current count value is retentive and will keep its current count after a loss of power to the PLR. In the example below, the counter will stop counting when it reaches the preset value of 20. Counter status bit C1 will be ON when the current value is 20.

Counter Mode 4 (Continuous Count, Retentive)

Mode 4 Counter operation is similar to Mode 2 except its current count value is retentive. Mode 4 Counter will count up to a fixed preset value and continue counting after the preset value. Additionally, the current count value is retentive and will keep its current count after a loss of power to the PLR. In the example below, the counter will continue counting after its preset value of 20. Counter status bit C1 will be ON when the current value is 20.
Counter Mode 5 (Continuous Count, Up-Down Counter, Non-Retentive)

Mode 5 Counter operation is similar to Mode 2 where its current count value is continuous and non-retentive, except its C1 status bit will only be ON when the counter counts up to its preset, or down to its preset from a count higher than its preset. Even with its direction bit set to ON, it will not turn on its C1 status bit when it counts down to zero. The C1 status bit is fixed to the non-zero preset value regardless of the state of the direction bit. Additionally, the Mode 5 counter is always reset to zero, unrelated to the state of its direction bit.

The Mode 5 Counter will count up to a fixed preset value and continue counting after the preset value. Additionally, the current count value is non-retentive and will reset to zero on a loss of power to the PLR. In the example below, the counter will continue counting after its preset value of 20. Counter status bit C1 will be ON when the current value is 20.
Counter Mode 6 (Continuous Count, Up-Down Counter, Retentive)

Mode 6 Counter operation is similar to Mode 4 where its current count value is continuous and retentive, except its C1 status bit will only be ON when the counter counts up to its preset or down to its preset from a count higher than its preset. Even with its direction bit set to ON, it will not turn on its C1 status bit when it counts down to zero. The C1 status bit is fixed to the non-zero preset value regardless of the state of the direction bit. Additionally, the Mode 5 counter is always reset to zero, unrelated to the state of its direction bit.

The Mode 6 Counter will count up to a fixed preset value and continue counting after the preset value. Additionally, the current count value is retentive and will keep its current count after a loss of power to the PLR. In the example below, the counter will continue counting after its preset value of 20. Counter status bit C1 will be ON when the current value is 20.
High Speed Counters (DC Version PLRs Only)

The DC powered version PLRs include two 1Khz high speed inputs on terminal I1 and I2. These can be used as general purpose DC inputs or can be wired to a high speed input device (encoder, etc.) when configured for high speed counting. These are often used for counting something moving very fast (>40hz) or used as a speed reference on a machine. The high speed counters are configured using the same software Edit Contact/Coil dialog box, except selecting Counter Mode 7 or Mode 8.

High Speed Counter Mode 7 (DC powered versions only)

The Mode 7 High Speed Counter can use either input terminals I1 or I2 for forward up-counting to 1Khz maximum at 24VDC high speed input signal. The selected Counter Coil (C1-CF) will turn ON when the pulse count reaches the target setpoint and remain ON. The counter will reset when the preceding rung is inactive or the Reset Input is active.

The figure to the right shows the relationship between the numbered block diagram for a Mode 7 Counter, the ladder diagram view, and the software Edit Contact/Coil dialog box.
High Speed Counter Mode 8 (DC powered versions only)

The Mode 8 High Speed Counter can use either input terminals I1 or I2 for forward up-counting to 1Khz maximum at 24VDC high speed input signal. The selected Counter Coil (C1-CF) will turn ON when the pulse count reaches the target “Preset ON” value and remain ON until the pulse count reaches the target “Preset OFF” value. The Fixed Time xxxx. The counter will reset when the preceding rung is inactive.

The table below describes each configuration parameter for High Speed Counter Mode 8:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Counting Mode (8) frequency comparison</td>
</tr>
<tr>
<td>2</td>
<td>High speed counting input terminal: I1 or I2 only</td>
</tr>
<tr>
<td>3</td>
<td>Use (I1 ~ gF) to RESET the counting value</td>
</tr>
<tr>
<td></td>
<td>ON: the counter reset to zero</td>
</tr>
<tr>
<td></td>
<td>OFF: the counter continues to count</td>
</tr>
<tr>
<td>4</td>
<td>Current Count Value, range:0~999999</td>
</tr>
<tr>
<td>5</td>
<td>Preset Value, range:0~999999</td>
</tr>
<tr>
<td>6</td>
<td>Counter Coil Number (C1 ~ CF total: 15 counters)</td>
</tr>
</tbody>
</table>

The figure to the right shows the relationship between the numbered block diagram for a Mode 8 Counter, the ladder diagram view, and the software Edit Contact/Coil dialog box.
Timer Instructions

The SG2 PLR includes a total of 15 separate timers that can be used throughout a program. Each timer has a choice of 7 operation modes, 6 for general purpose timing and 1 (mode 7) for a pulse timer. Additionally, each timer has 6 parameters for proper configuration. The table below describes each configuration parameter and lists each compatible memory type for configuring counters.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Compatible Instructions</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>/L55443</td>
<td>Timer Mode (1-7)</td>
<td>Inputs</td>
<td>I1-Ic/i1-iC</td>
</tr>
<tr>
<td>/L55444</td>
<td>Timer Unit: 0.00 - 99.99 sec</td>
<td>Outputs</td>
<td>Q1-Q8/q1-q8</td>
</tr>
<tr>
<td>/L55445</td>
<td>ON: the timer reset to zero</td>
<td>Auxiliary coil</td>
<td>M1-MF/m1-mF</td>
</tr>
<tr>
<td>/L55446</td>
<td>OFF: the timer continues to time</td>
<td>Expansion inputs</td>
<td>X1-Xc/x1-xC</td>
</tr>
<tr>
<td>/L55447</td>
<td>Current timer value</td>
<td>Expansion outputs</td>
<td>Y1-Yc/y1-yC</td>
</tr>
<tr>
<td>/L55448</td>
<td>Timer preset value</td>
<td>RTC</td>
<td>R1-RF/r1-rF</td>
</tr>
<tr>
<td></td>
<td>Timer preset value</td>
<td>Counter</td>
<td>C1-CF/c1-cF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Timer</td>
<td>T1-TF/t1-TF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Analog comparator</td>
<td>G1-GF/g1-gF</td>
</tr>
</tbody>
</table>

Timer Mode 1 (ON-Delay)

Mode 1 Timer (ON-Delay) will time up to a fixed preset value and stop timing when the current time is equal to the preset value. Additionally, the current time value is non-retentive and will reset to zero on a loss of power to the PLR. In the example below, the timer will stop timing when it reaches the preset value of 5 seconds. Timer status bit T1 will be ON when the current value is 5.
Timer Mode 2 (ON-Delay with Reset)

Mode 2 Timer is an ON-Delay with reset that will time up to a fixed preset value and stop timing when the current time is equal to the preset value. Additionally, the current time value is non-retentive and will reset to zero on a loss of power to the PLR. The timer reset input is Input I1. In the example below, the timer will stop timing when it reaches the preset value of 5 seconds. Timer status bit T1 will be ON when the current value is 5.
**Timer Mode 3 (OFF-Delay)**

Mode 3 Timer is an OFF-Delay with reset that will time up to a fixed preset value and stop timing when the current time is equal to the preset value. Additionally, the current time value is non-retentive and will reset to zero on a loss of power to the PLR. In the example below, the timer reset input is Input I1. Also in the example below, timer status bit T1 will be ON immediately when its rung is true. The timer will only begin timing up when its rung changes to false. Timer status bit T1 will turn OFF when the current time value reaches 10 seconds.

![Diagram of Timer Mode 3](image)
Timer Mode 4 (OFF-Delay)

Mode 4 Timer is an OFF-Delay with reset that will time up to a fixed preset value and stop timing when the current time is equal to the preset value. Additionally, the current time value is non-retentive and will reset to zero on a loss of power to the PLR. In the example below, the timer reset input is Input I1. Also in the example below, the timer status bit T1 will turn ON only after its rung transitions from true to false. Timer status bit T1 will turn OFF when the current time value reaches 10 seconds.
Timer Mode 5 (FLASH without Reset)

Mode 5 Timer is a Flash timer without reset that will time up to a fixed preset value then change the state of its status bit when the current time is equal to the preset value. Additionally, the current time value is non-retentive and will reset to zero on a loss of power to the PLR. In the example below, timer status bit T1 will be ON immediately when its rung is true and begin its timing sequence. Timer status bit T1 will turn OFF when the current time value reaches its preset of 10 seconds. This Flash sequence of the timer status bit T1 will continue as long as its rung remains true.
Timer Mode 6 (FLASH with Reset)

Mode 6 Timer is a Flash timer without reset that will time up to a fixed preset value then change the state of its status bit when the current time is equal to the preset value. Additionally, the current time value is non-retentive and will reset to zero on a loss of power to the PLR. In the example below, the timer reset input is Input I1. Also in the example below, timer status bit T1 will be ON immediately when its rung is true and begin its timing sequence. Timer status bit T1 will turn OFF when the current time value reaches its preset of 5 seconds. This Flash sequence of the timer status bit T1 will continue as long as its rung remains true.
Timer Mode 7 (FLASH Cascade without Reset)

Mode 7 Timer is a Flash timer without reset that uses two timers in a cascade configuration. The cascade configuration connects the timer status bit of first timer to enable the second timer. The second timer will time up to its preset value then flash and its timer status bit will enable the first timer. Additionally, the current time value is non-retentive and will reset to zero on a loss of power to the PLR.

In the example below, timer status bit T1 will be ON after it completes its timing sequence of 2.5 seconds. Timer 2 will then begin its timing sequence of 1 second. When the current time value of Timer 2 reaches its preset of 1 second, its status bit T2 will flash and Timer 1 will begin timing again.

This type of cascade timer is often used in combination with a counter in applications where it is necessary to count the number of time cycles completed.

Note: Timer Mode 7 uses two timers. These timers cannot be reused as timers for other modes in other areas of the program.
Real Time Clock (RTC) Instructions

The SG2 PLR includes a total of 15 separate RTC instructions that can be used throughout a program. Each RTC instruction has a choice of 3 operation modes, and has 10 parameters for proper configuration. The initial clock/calendar setting for each connected SG2 is set using the Operation»RTC Set menu selection from the SG2 Client software.

Daily Mode 1

The Daily Mode 1 allows the Rx coil to activate based on a fixed time across a defined set of days per week. The configuration dialog below allows for selection of the number of days per week (i.e. Mon-Fri) and the Day and Time for the Rx coil to activate ON, and Day and Time for the Rx coil to deactivate OFF.

Interval Time Mode 2 (weekly)

The Interval Time Mode 2 allows the Rx coil to activate based on time and day per week. The configuration dialog below allows for selection of Day and Time for the Rx coil to activate ON, and Day and Time for the Rx coil to deactivate OFF.
Year-Month-Day Mode 3

The Year-Month-Day Mode 3 allows the Rx coil to activate based on Year, Month, and Date. The configuration dialog below allows for selection of Year and Date for the Rx coil to activate ON, and Year and Date for the Rx coil to deactivate OFF.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RTC mode 3, Year-Month-Day</td>
</tr>
<tr>
<td>2</td>
<td>Set RTC Year ON</td>
</tr>
<tr>
<td>3</td>
<td>Set RTC Year OFF</td>
</tr>
<tr>
<td>4</td>
<td>Displays RTC current time: Year-Month-Day</td>
</tr>
<tr>
<td>5</td>
<td>Set RTC Month ON</td>
</tr>
<tr>
<td>6</td>
<td>Set RTC Day ON</td>
</tr>
<tr>
<td>7</td>
<td>Set RTC Month OFF</td>
</tr>
<tr>
<td>8</td>
<td>Set RTC Day OFF</td>
</tr>
<tr>
<td>9</td>
<td>RTC coil number (R1-RF Total: 15 RTCs)</td>
</tr>
</tbody>
</table>
Comparator Instructions

The SG2 PLR includes a total of 15 separate comparator instructions that can be used throughout a program. Each comparator has a choice of 5 operation modes. Additionally, each comparator has 7 parameters for proper configuration. The table below describes each configuration parameter, and lists each compatible memory type for configuring counters.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Comparison Mode(1~5)</td>
</tr>
<tr>
<td>2</td>
<td>Ax analog input (A1~A8), the present value of the timer, counter, or a constant value.</td>
</tr>
<tr>
<td>3</td>
<td>Ay analog input (A1~A8), the present value of the timer, counter, or a constant value.</td>
</tr>
<tr>
<td>4</td>
<td>Ax analog input value(0.00~9.99)</td>
</tr>
<tr>
<td>5</td>
<td>Ay analog input value (0.00~9.99)</td>
</tr>
<tr>
<td>6</td>
<td>Set reference comparative value: could be constant, or the present value of the timer, counter and analog input.</td>
</tr>
<tr>
<td>7</td>
<td>Output terminal(G1~GF)</td>
</tr>
</tbody>
</table>

Comparison Mode 1 \((A_y - 6 \leq A_x \leq A_y + 6, 7 \text{ ON})\)

Comparison Mode 2 \((A_x \leq A_y, 7 \text{ ON})\)

Comparison Mode 3 \((A_x \geq A_y, 7 \text{ ON})\)

Comparison Mode 4 \((A_x \leq 6, 7 \text{ ON})\)

Comparison Mode 5 \((A_x \geq 6, 7 \text{ ON})\)

Example 1: Analog Signal Compare

In the example below, Mode 4 is the selected function that compares the value of analog input A1 to a constant value (N) of 2.50. Status coil G1 turns ON is A1 is <= to 2.50.

![Example Diagram]
Example 2: Timer/Counter Preset Value Compare

The Comparator instruction can be used to compare Timer, Counter, and RTC values to a constant value or to each other. In this example below, Mode 5 is the selected function that compares the value of Counter (C1) to a constant value (N) of 15 counts (the decimal point is ignored). Status coil G1 turns ON if C1 is ≥ to 15 counts.
HMI Display Instructions

The SG2 PLR includes a total of 15 HMI instructions that can be used throughout a program. Each HMI instruction can be configured to display information on the SG2 12×4 character LCD in text, numeric, or bit format for items such as current value and target value for timers/counters, Input/Output bit status, RTC (real time clock) and Analog comparator. Each HMI instruction is configured separately using the Edit»HMI/Text menu selection from the SG2 Client software.

In the adjacent example, HMI instruction H1 is configured to display the value of I1 and T1, and some descriptive text. Numeric display data selections are Timer, Counter, RTC, and Analog. Bit display data selections for “ON” and “OFF” messages are “I” inputs, “M” internal relays, and “X” expansion inputs.

- Allows the SEL button on the SG2 keypad to activate the selected message onto the LCD even when Hx coil is inactive.
- Allows the HMI message to include coil number and selected value (i.e. T1=003 Sec).
- Provides access to the Analog Display Set dialog for scaling and offset parameters shown below.

The Analog Display Set dialog allows the user to specify a scaling factor (Gain) and an offset for each analog input value.

A phone number can be displayed on the screen to alert an operator to call for help.

Note: The Phone Number field does not dial a modem or allow for a modem connection.
Each HMI instruction has a choice of 2 operation modes. The table below describes each configuration parameter.

### PWM Output Instruction (DC Transistor Output Models Only)

The transistor output model SG2 PLRs include the capability to provide a PWM (Pulse Width Modulation) output on terminal Q1. The PWM instruction is able to output up to an 8-stage PWM waveform.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Display Mode (1-2)</td>
</tr>
<tr>
<td>2</td>
<td>Display the present stage as operation (0-8)</td>
</tr>
<tr>
<td>3</td>
<td>Input Selected Stage 1 (I1-gF)</td>
</tr>
<tr>
<td>4</td>
<td>Input Selected Stage 2 (I1-gF)</td>
</tr>
<tr>
<td>5</td>
<td>Input Selected Stage 3 (I1-gF)</td>
</tr>
<tr>
<td>6</td>
<td>Set PWM pulse width (0-32768ms)</td>
</tr>
<tr>
<td>7</td>
<td>Set PWM Period (1-32768ms)</td>
</tr>
<tr>
<td>8</td>
<td>PWM output terminal P1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enable</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>8 Output PWM</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>OFF</td>
</tr>
<tr>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>1</td>
<td>Set stage 1</td>
</tr>
<tr>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
<td>ON</td>
<td>2</td>
<td>Set stage 2</td>
</tr>
<tr>
<td>ON</td>
<td>OFF</td>
<td>ON</td>
<td>OFF</td>
<td>3</td>
<td>Set stage 3</td>
</tr>
<tr>
<td>ON</td>
<td>OFF</td>
<td>ON</td>
<td>ON</td>
<td>4</td>
<td>Set stage 4</td>
</tr>
<tr>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
<td>5</td>
<td>Set stage 5</td>
</tr>
<tr>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
<td>6</td>
<td>Set stage 6</td>
</tr>
<tr>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>7</td>
<td>Set stage 7</td>
</tr>
<tr>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>8</td>
<td>Set stage 8</td>
</tr>
</tbody>
</table>
Data Link/Remote I/O Instruction (SG2-20Vxxx Models Only)

The SG2-20Vxxx transistor output models include the capability to link additional SG2-20Vxx units via the RS-485 connection terminals. Up to 8 additional SG2 units can be configured as independent Slave nodes, each running their own logic program and their I/O linked to one Master PLR. Up to 2 additional SG2 units can be configured as Remote I/O nodes, and linked to one Master PLR.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mode setting (1, 2) 1: sending 2: receiving</td>
</tr>
<tr>
<td>2</td>
<td>Set the send/receive points (1-8)</td>
</tr>
<tr>
<td>3</td>
<td>Set the send/receive points</td>
</tr>
<tr>
<td>4</td>
<td>Send/receive memory list location</td>
</tr>
<tr>
<td>5</td>
<td>I/O link output terminal (L1-L8)</td>
</tr>
</tbody>
</table>

Note: Only one “Mode 1 Send” Data Link instruction (L1-L8) is allowed per Master PLR. All other Data Link instructions must be “Mode 2 Receive” instructions.

The Mode 2 Receive memory range is determined by the Controller ID. Each controller ID is allocated a range of 8 I/O points (Wx-Wx) that can be read into the Master PLR using a DataLink instruction. The adjacent table shows the memory range of Wx locations associated with each controller ID.

<table>
<thead>
<tr>
<th>ID</th>
<th>Memory List Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>W1~W8</td>
</tr>
<tr>
<td>1</td>
<td>W9~W16</td>
</tr>
<tr>
<td>2</td>
<td>W17~W24</td>
</tr>
<tr>
<td>3</td>
<td>W25~W32</td>
</tr>
<tr>
<td>4</td>
<td>W33~W40</td>
</tr>
<tr>
<td>5</td>
<td>W41~W48</td>
</tr>
<tr>
<td>6</td>
<td>W49~W56</td>
</tr>
<tr>
<td>7</td>
<td>W57~W64</td>
</tr>
</tbody>
</table>

The Data Link instruction below is setup for Mode 1 Send where the Master PLR is sending 5 I/O points of Inputs to each connected Slave PLR. The starting Input is I03 with the resulting range of 5 sending inputs equal to I3 – I7.
**Example 1: Data Link Mode 1**

Set $\text{①} = 1$, $\text{②} = 5$, set $\text{③}$ as the initiate of I3, the state of actual sending terminal I3~I7 is sent to memory list; the controller ID = 3, the state of corresponding memory list position W17~W24-④ and relationship of sending terminal is as below:

$$
\begin{align*}
\text{①} &= 1 & \text{②} &= 5 & \text{③} : & I3 \sim I7 & \text{ID} = 3 & (\text{④} : W17 \sim W24)
\end{align*}
$$

<table>
<thead>
<tr>
<th>Memory List Position</th>
<th>W17</th>
<th>W18</th>
<th>W19</th>
<th>W20</th>
<th>W21</th>
<th>W22</th>
<th>W23</th>
<th>W24</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corresponding receiving and sending terminal</td>
<td>I3</td>
<td>I4</td>
<td>I5</td>
<td>I6</td>
<td>I7</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Example 2: Data Link Receive mode 2**

Set $\text{①} = 1$, $\text{②} = 5$, set $\text{③}$ as start from I3, set $\text{④}$ as start from W17, when enabling the Data Link, the state 'ON/OFF' of I3~I7 is controlled by the state of memory list position W17~W21-④, which is irrelative to the actual state of input terminal.

$$
\begin{align*}
\text{①} &= 1 & \text{②} &= 5 & \text{③} : & I3 \sim I7 & \text{④} : W17 \sim W21
\end{align*}
$$

<table>
<thead>
<tr>
<th>Memory List Position:</th>
<th>W17</th>
<th>W18</th>
<th>W19</th>
<th>W20</th>
<th>W21</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corresponding Receiving and Sending Terminal:</td>
<td>I3</td>
<td>I4</td>
<td>I5</td>
<td>I6</td>
<td>I7</td>
</tr>
</tbody>
</table>
Chapter 5: Function Block Diagram Programming

FBD Instructions

### Coil Block Instruction

```
   I
   N O P
   Q 0 1
   Q
```

### Logic Block Instructions

```
   B05
   AND
   B 0 1
   N O P
   N O P
   N O P
```

**Function Block**

<table>
<thead>
<tr>
<th>Function Block</th>
<th>1 Input Terminal</th>
<th>2 Output Coil</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>I</td>
<td>I01~I0C(12)</td>
<td></td>
</tr>
<tr>
<td>Expansion Input</td>
<td>X</td>
<td>X01~X0C(12)</td>
<td></td>
</tr>
<tr>
<td>Output</td>
<td>Q</td>
<td>Q01~Q08(8)</td>
<td></td>
</tr>
<tr>
<td>Expansion Output</td>
<td>Y</td>
<td>Y01~Y0C(12)</td>
<td></td>
</tr>
<tr>
<td>Auxiliary</td>
<td>M</td>
<td>M01~M0F(15)</td>
<td></td>
</tr>
<tr>
<td>Knob</td>
<td>N</td>
<td>N01~N0F(15)</td>
<td></td>
</tr>
<tr>
<td>HMI</td>
<td>H</td>
<td>H01~H0F(15)</td>
<td></td>
</tr>
<tr>
<td>PWM</td>
<td>P</td>
<td>P01(1)</td>
<td></td>
</tr>
<tr>
<td>SHIFT</td>
<td>S</td>
<td>S01(1)</td>
<td></td>
</tr>
<tr>
<td>I/O LINK</td>
<td>L</td>
<td>L01~L08(8)</td>
<td></td>
</tr>
<tr>
<td>Logic/Function Block</td>
<td>B</td>
<td>B01~B99(99)</td>
<td></td>
</tr>
<tr>
<td>Normal ON</td>
<td>Hi</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Normal OFF</td>
<td>Lo</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No Connection</td>
<td>Nop</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
PWM Function Block

The PWM output terminal 'Q1' can output 8 PWM waveforms. (only provided for transistor output version)

SHIFT Function Block

Input terminal description:

Enable input →
Shift input →

1. Shift code
2. Shift type
3. Shift number

Symbol Description:
① SHIFT code (Total 1 group)
② Set output type (Q, Y)
③ Set output shift number (1-8)

Symbol: Q
Shift output range: Q1~Q5

Enable
Shift input

Q1
Q2
Q3
Q4
Q5
**AND Logic Block**

**FBD**

\[
\begin{array}{c}
\text{I 0 1} \quad \text{AND} \\
\text{I 0 2} \\
\text{I 0 3}
\end{array}
\]

\[
B \times x
\]

**Ladder**

\[
\begin{array}{c}
\text{I 0 1} \\
\text{I 0 2} \\
\text{I 0 3}
\end{array}
\]

Note: The input terminal is NOP which is equivalent to ‘Hi’

**AND (Edge) Logic Block**

**FBD**

\[
\begin{array}{c}
\text{I 0 1} \quad \text{AND} \\
\text{I 0 2} \\
\text{I 0 3}
\end{array}
\]

\[
B \times x
\]

**Ladder**

\[
\begin{array}{c}
\text{I 0 1} \\
\text{I 0 2} \\
\text{I 0 3}
\end{array}
\]

**NAND Logic Block**

**FBD**

\[
\begin{array}{c}
\text{I 0 1} \quad \text{NAND} \\
\text{I 0 2} \\
\text{I 0 3}
\end{array}
\]

\[
B \times x
\]

**Ladder**

\[
\begin{array}{c}
i 0 1 \\
i 0 2 \\
i 0 3
\end{array}
\]

**NAND (Edge) Logic Block**

**FBD**

\[
\begin{array}{c}
\text{I 0 1} \quad \text{NAND} \\
\text{I 0 2} \\
\text{I 0 3}
\end{array}
\]

\[
B \times x
\]

**Ladder**

\[
\begin{array}{c}
i 0 1 \\
i 0 2 \\
i 0 3
\end{array}
\]

Note: The input terminal is NOP which is equivalent to ‘Hi’
OR Logic

FBD

\[
\begin{array}{c}
\text{I 0 1} \quad \text{OR} \\
\text{I 0 2} \\
\text{I 0 3}
\end{array}
\]

Ladder

\[
\begin{array}{c}
\text{I01} \\
\text{I02} \\
\text{I03}
\end{array}
\]

NOR Logic

FBD

\[
\begin{array}{c}
\text{I 0 1} \quad \text{NOR} \\
\text{I 0 2} \\
\text{I 0 3}
\end{array}
\]

Ladder

\[
\begin{array}{c}
i01 \\
i02 \\
i03
\end{array}
\]

Note: The input terminal is NOP which is equivalent to ‘Lo’

XOR Logic

FBD

\[
\begin{array}{c}
\text{I 0 1} \quad \text{XOR} \\
\text{I 0 2} \\
\text{I 0 3}
\end{array}
\]

Ladder

\[
\begin{array}{c}
i01 \\
i02 \\
i01 \quad i02
\end{array}
\]

SR Logic Block

FBD

\[
\begin{array}{c}
\text{I 0 1} \quad \text{SR} \\
\text{I 0 2} \\
\text{I 0 3}
\end{array}
\]

Ladder

\[
\begin{array}{c}
i01 \\
i02 \\
i01 \quad \text{XX}
\end{array}
\]
**NOT Logic Block**

FBD

```
\[ \begin{array}{c}
\text{I} \quad \text{0} \quad \text{1} \\
\text{NOT} \\
\text{B} \quad \text{x} \\
\text{y} \quad \text{y}
\end{array} \]
```

Ladder

```
\[ \begin{array}{c}
i01 \\
\text{xx} \\
\text{I} \quad \text{NOT}
\end{array} \]
```

Note: The input terminal is NOP which is equivalent to ‘Hi’

**Pulse Logic Block**

FBD

```
\[ \begin{array}{c}
\text{I} \quad \text{0} \quad \text{1} \\
\text{P} \\
\text{B} \quad \text{x} \\
\text{y} \quad \text{y}
\end{array} \]
```

Ladder

```
\[ \begin{array}{c}
i01 \\
\text{xx} \\
\text{I} \quad \text{P}
\end{array} \]
```

Note: The input terminal is NOP which is equivalent to ‘Lo’

**Counter Function Block**

**Counter Mode 1**

Counting Input →

Up/Down Counting →

Reset →

Counting Parameter →

```
\[ \begin{array}{c}
\text{Counting Input} \\
\text{Up/Down Counting} \\
\text{Reset} \\
\text{Counting Parameter}
\end{array} \]
```

```
\[ \begin{array}{c}
\text{C} \\
\text{n} \quad \text{t} \\
\text{B} \quad \text{x} \\
\text{y} \quad \text{y}
\end{array} \]
```

Note: The “>” means the current value appeared will be greater than present value.

**Counter Mode 2**

Counting Input →

Up/Down Counting →

Reset →

Counting Parameter →

```
\[ \begin{array}{c}
\text{Counting Input} \\
\text{Up/Down Counting} \\
\text{Reset} \\
\text{Counting Parameter}
\end{array} \]
```

```
\[ \begin{array}{c}
\text{C} \\
\text{n} \quad \text{t} \\
\text{B} \quad \text{x} \\
\text{y} \quad \text{y}
\end{array} \]
```

Note: The “>” means the current value appeared will be greater than present value.
Counter Mode 3

Counting Input → \( C \times \) B ∨ x
Up/Down Counting → \( \uparrow / \downarrow \) y
Reset → \( R e s \) B ∨ y
Counting Parameter → \( P a r \) PD C y

Note: The "PD" means the current value will be retain until the power recover.

Counter Mode 4

Counting Input → \( C \times \) B ∨ x
Up/Down Counting → \( \uparrow / \downarrow \) y
Reset → \( R e s \) B ∨ y
Counting Parameter → \( P a r \) PD C y

Counter Mode 5

Counting Input → \( C \times \) B ∨ x
Up/Down Counting → \( \uparrow / \downarrow \) y
Reset → \( R e s \) B ∨ y
Counting Parameter → \( P a r \) C y

Counter Mode 6

Counting Input → \( C \times \) B ∨ x
Up/Down Counting → \( \uparrow / \downarrow \) y
Reset → \( R e s \) B ∨ y
Counting Parameter → \( P a r \) C y PD

Note: The "C" means that will keep the current value in 0 during the Reset pin be enable.
High Speed Counter Function Block

Counter Mode 7

Counting Input →
Up/Down Counting →
Reset →
Counting Parameter →

Counter Mode 8

Counting Input →
Up/Down Counting →
Reset →
Counting Parameter →

Note: High speed input terminal I1, I2

Timer Function Block

Timer Mode 1 (ON-Delay A Mode)

Enable Input →
Timing Parameter →

Timer Mode 2 (ON-Delay B Mode)

Enable Input →
Reset →
Timing Parameter →
RTC Function Block

RTC Mode 1 (Daily)

Enable Input →

RTC Parameter →

RTC Mode 2 (Weekly)

Enable Input →

RTC Parameter →

RTC Mode 3 (Year Month Day)

Enable Input →

RTC Parameter →
Analog comparator Function Block

Analog Comparison Mode 1

Enable Input → \( E_n \) \( \rightarrow \) \( B \times x \)
Analog Input → \( A \times \) \( \nrightarrow \) \( A_y \) \( \rightarrow \) \( R \)
Analog Input → \( A_y \) \( \nrightarrow \) \( \leq A_x \) \( \leq \) \( B \times y \)
Reference → \( \text{Ref} \) \( \nrightarrow \) \( A_y \) \( \rightarrow \) \( R \)

Analog Comparison Mode 2

Enable Input → \( E_n \) \( \rightarrow \) \( B \times x \)
Analog Input → \( A \times \) \( \nrightarrow \) \( A_y \) \( \rightarrow \) \( \text{Ref} \)
Analog Input → \( A_y \) \( \nrightarrow \) \( \leq A_y \rightarrow B \times y \)
Reference → \( \text{Ref} \) \( \nrightarrow \) \( \) \( \) \n
Analog Comparison Mode 3

Enable Input → \( E_n \) \( \rightarrow \) \( B \times x \)
Analog Input → \( A \times \) \( \nrightarrow \) \( A_y \) \( \rightarrow \) \( \text{Ref} \)
Analog Input → \( A_y \) \( \nrightarrow \) \( \geq A_y \rightarrow B \times y \)
Reference → \( \text{Ref} \) \( \nrightarrow \) \( \) \( \) \n
Analog Comparison Mode 4

Enable Input → \( E_n \) \( \rightarrow \) \( B \times x \)
Analog Input → \( A \times \) \( \nrightarrow \) \( \text{Ref} \)
Analog Input → \( \text{Ref} \) \( \nrightarrow \) \( \geq A_x \rightarrow B \times y \)
Reference → \( \text{Ref} \) \( \nrightarrow \) \( \) \( \) \n
Analog Comparison Mode 5

Enable Input → \( E_n \) \( \rightarrow \) \( B \times x \)
Analog Input → \( A \times \) \( \nrightarrow \) \( \text{Ref} \)
Analog Input → \( \text{Ref} \) \( \nrightarrow \) \( \leq A_x \rightarrow B \times y \)
Reference → \( \text{Ref} \) \( \nrightarrow \) \( \) \( \)