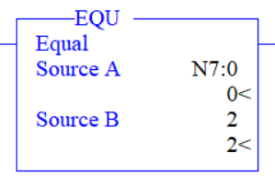
Title: **Comparison Functions** Handout: 7

Course: Introduction to Automation Unit: Introduction of PLC CLO: 4

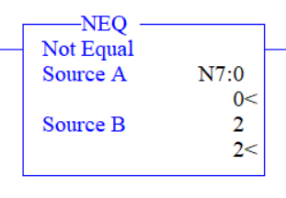
**Objectives**

1. Student shall develop a knowledge of the Equal (EQU) and Not Equal (NEQ) comparison instruction.
2. Student shall develop a knowledge of the Greater Than (GRT) and Greater Than or Equal (GEQ) comparison instruction.
3. Student shall develop a knowledge of the Less Than (LES) and Less Than or Equal (LEQ) comparison instruction.

**Theory**

Equal (EQU)

This instruction is a comparison instruction that evaluates two numbers to determine if they are the same value. This instruction will gain logical continuity when *Source A* equals *Source B*. If *Source A* does not equal *Source B*, this instruction will not have logical continuity. This instruction requires that an output instruction exists in the rung. See Note A below.

Not Equal (NEQ)

This instruction is a comparison instruction that evaluates two numbers to determine if they are not the same value. This instruction will gain logical continuity when *Source A* does not equal *Source B*. If *Source A* equals *Source B*, this instruction will not have logical continuity. This instruction requires that an output instruction exists in the rung. See Note A below.

**Note A:** Source A and Source B can either be constant values or addresses that contain values, however Source A and Source B cannot both be constants.

**Graphics**

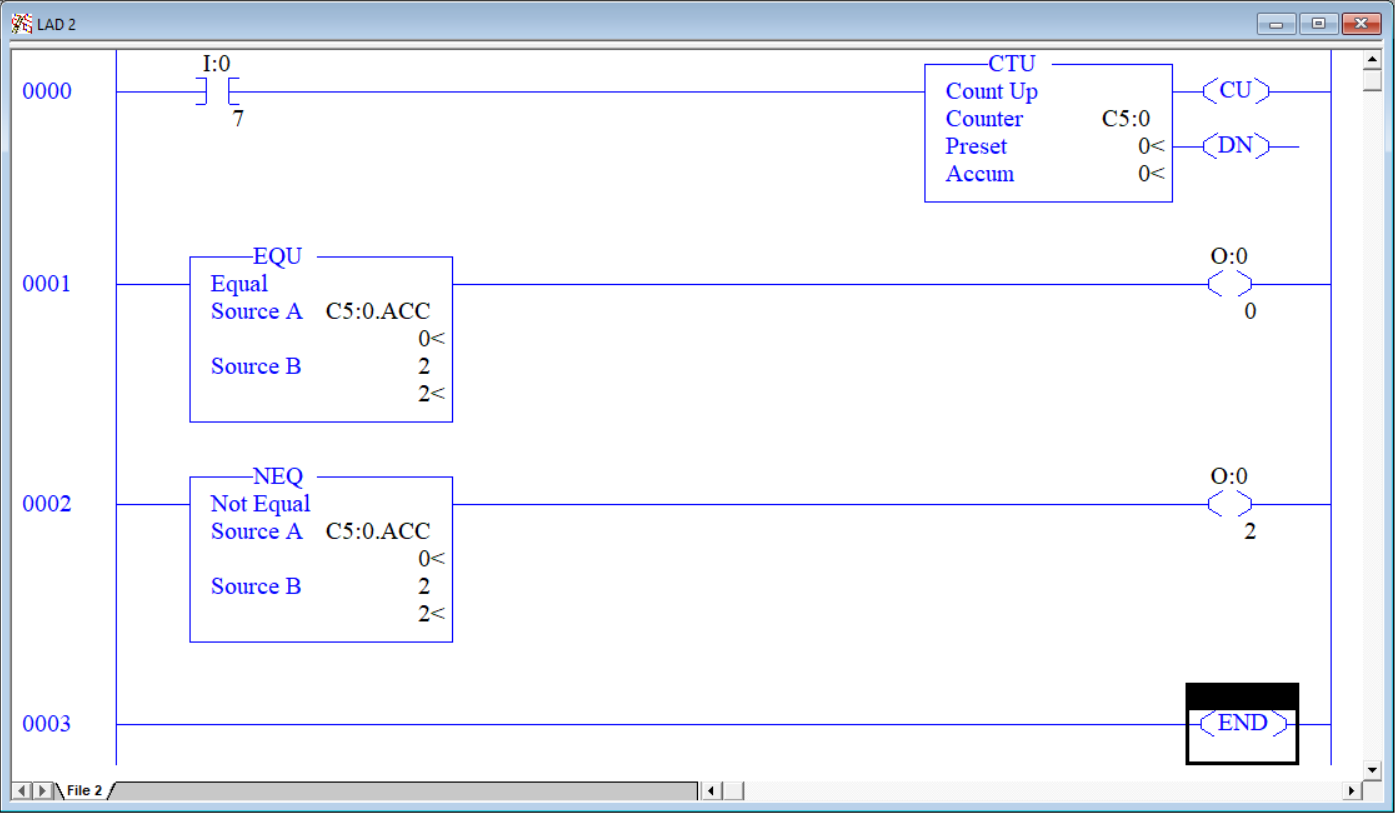
|  |  |
| --- | --- |
| EQU | NEQ |
|  |  |

**Devices**

|  |  |  |
| --- | --- | --- |
| Inputs | | |
| *Device* | *Description* | *Symbol* |
| NO Pushbutton (PB2) | Counter | CNT |
| Outputs | | |
| *Device* | *Description* | *Symbol* |
| Green Pilot Light | EQU Function Result | EQU\_LT |
| Red Pilot Light | NEQ Function Result | NOT\_EQU\_LT |

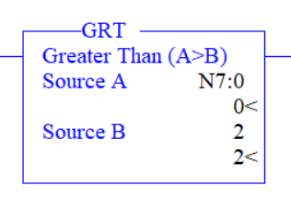
**Instructions**

Program the following logic.

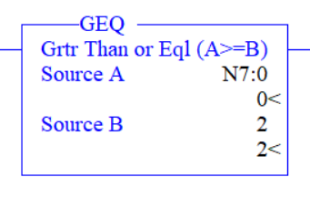


1. After downloading the program, notice that with C5:0.ACC = 0, the red pilot light is on since 0 ≠ 2.
2. Press PB2. Notice that with C5:0.ACC = 1, the red pilot light is on since 1 ≠ 2.
3. Press PB2. Notice that with C5:0.ACC = 2, the red pilot light is off and the green light is on since 2 = 2.

**Theory**

Greater Than (GRT)

This instruction is a comparison instruction that evaluates two numbers to determine if one is greater than the other. This instruction will gain logical continuity when *Source A* is greater than *Source B*. If *Source A* is less than or equal to *Source B*, this instruction will not have logical continuity. This instruction requires that an output instruction exists in the rung. See Note A below.

Greater Than or Equal (GEQ)

This instruction is a comparison instruction that evaluates two numbers to determine if one is at least the same value as another. This instruction will gain logical continuity when *Source A* is greater than or equal to *Source B*. If *Source A* is less than *Source B*, this instruction will not have logical continuity. This instruction requires that an output instruction exists in the rung. See Note A below.

**Note A:** Source A and Source B can either be constant values or addresses that contain values, however Source A and Source B cannot both be constants.

**Graphics**

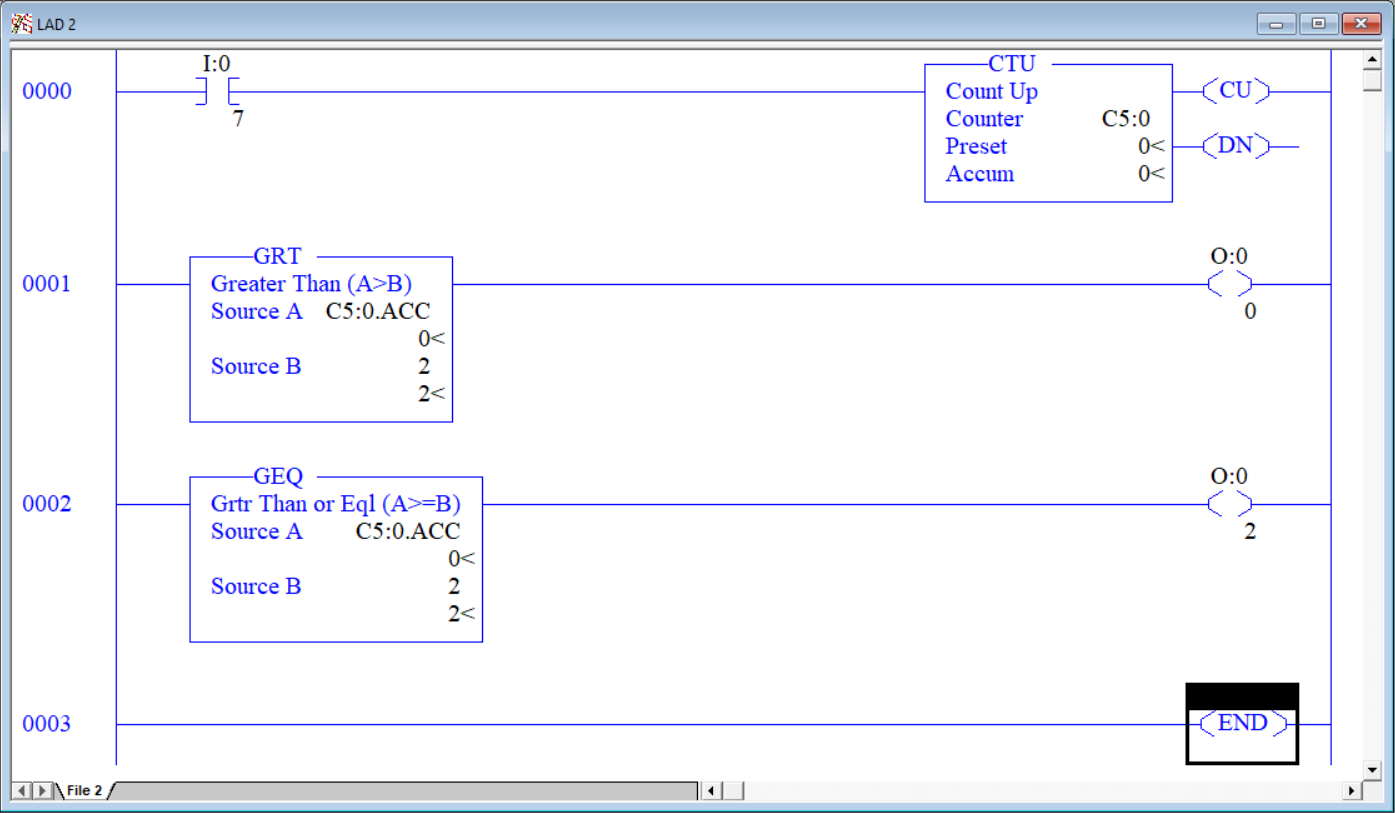
|  |  |
| --- | --- |
| GRT | GEQ |
|  |  |

**Devices**

|  |  |  |
| --- | --- | --- |
| Inputs | | |
| *Device* | *Description* | *Symbol* |
| NO Pushbutton (PB2) | Counter | CNT |
| Outputs | | |
| *Device* | *Description* | *Symbol* |
| Green Pilot Light | GRT Function Result | GRT\_LT |
| Red Pilot Light | GEQ Function Result | GEQ\_LT |

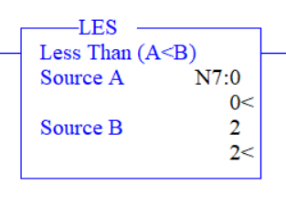
**Instructions**

Modify the program as shown below.

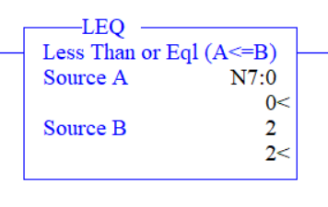


1. After downloading the program, notice that with C5:0.ACC = 0, the red and green pilot lights are off since 0 < 2.
2. Press PB2. With C5:0.ACC = 1 the red and green pilot lights stay off since 1 < 2.
3. Press PB2. With C5:0.ACC = 2, the red pilot light comes on since 2 = 2.
4. Press PB2. With C5:0.ACC = 3, the red and green pilot lights are on since 3 > 2.

**Theory**

Less Than (LES)

This instruction is a comparison instruction that evaluates two numbers to determine if one is less than the other. This instruction will gain logical continuity when *Source A* is less than *Source B*. If *Source A* is greater than or equal to *Source B*, this instruction will not have logical continuity. This instruction requires that an output instruction exists in the rung. See Note A below.

Less Than or Equal (LEQ)

This instruction is a comparison instruction that evaluates two numbers to determine if one is at most the same value as another. This instruction will gain logical continuity when *Source A* is less than or equal to *Source B*. If *Source A* is greater than *Source B*, this instruction will not have logical continuity. This instruction requires that an output instruction exists in the rung. See Note A below.

**Note A:** Source A and Source B can either be constant values or addresses that contain values, however Source A and Source B cannot both be constants.

**Graphics**

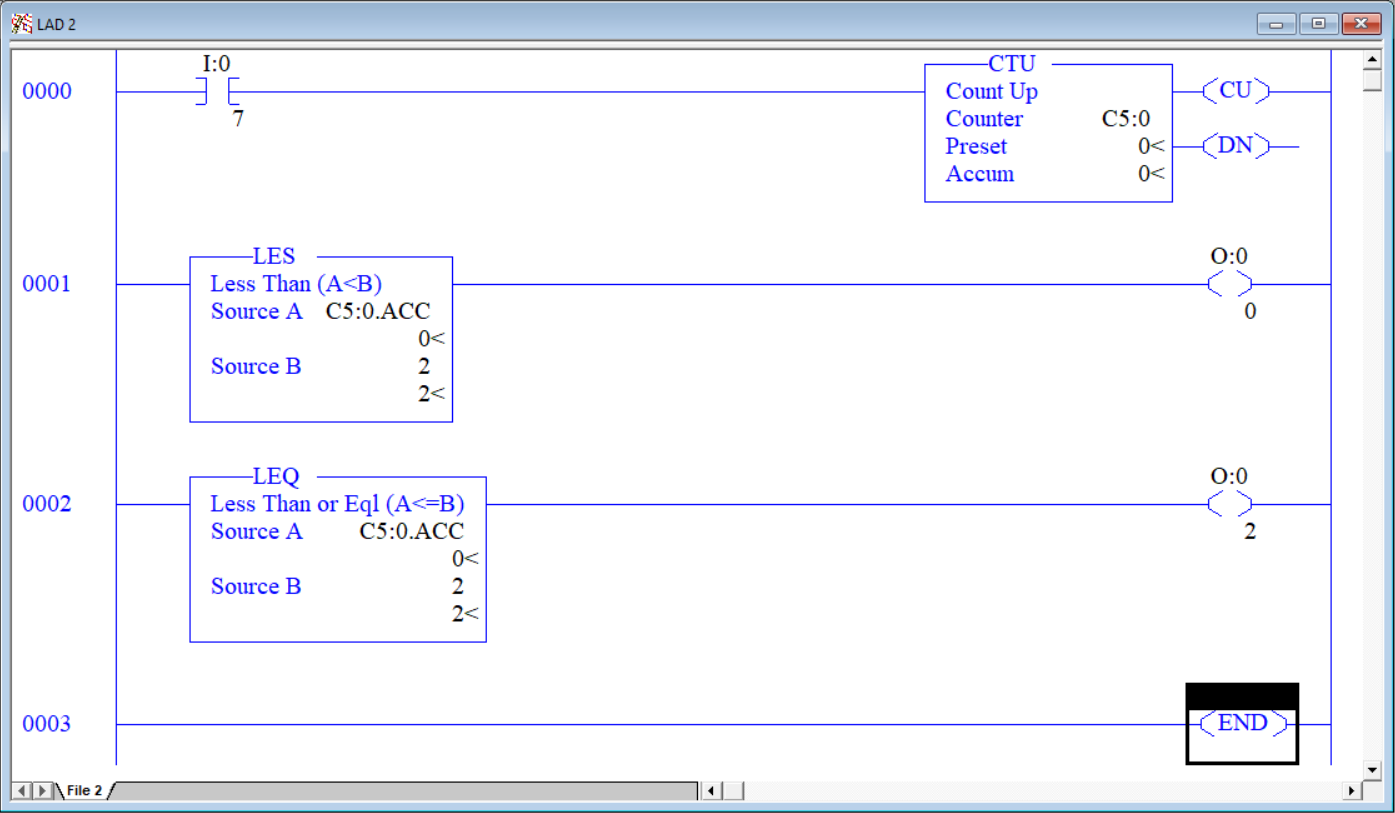
|  |  |
| --- | --- |
| LES | LEQ |
|  |  |

**Devices**

|  |  |  |
| --- | --- | --- |
| Inputs | | |
| *Device* | *Description* | *Symbol* |
| NO Pushbutton (PB2) | Counter | CNT |
| Outputs | | |
| *Device* | *Description* | *Symbol* |
| Green Pilot Light | LES Function Result | LES\_LT |
| Red Pilot Light | LEQ Function Result | LEQ\_LT |

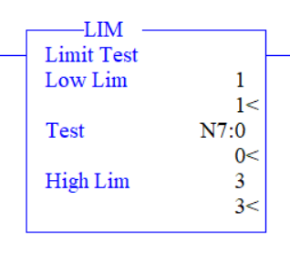
**Instructions**

Modify the program as shown below.



1. After downloading the program, notice that with C5:0.ACC = 0, the red and green pilot lights are on since 0 < 2.
2. Press PB2. With C5:0.ACC = 1 the red and green pilot lights stay on since 1 < 2.
3. Press PB2. With C5:0.ACC = 2, the green pilot light goes off since 2 = 2.
4. Press PB2. With C5:0.ACC = 3, the red and green pilot lights off since 3 > 2.

**Theory**

Limit (LIM)

This instruction is a comparison instruction that evaluates three numbers to determine if a given value is between two other values. This instruction will gain logical continuity when *Test* is greater than or equal to the *Low Lim* and is less than or equal to the *High Lim*. This instruction requires that an output instruction exists in the rung. See Note A below.

**Note A:** Source A and Source B can either be constant values or addresses that contain values, however Source A and Source B cannot both be constants.

**Graphics**

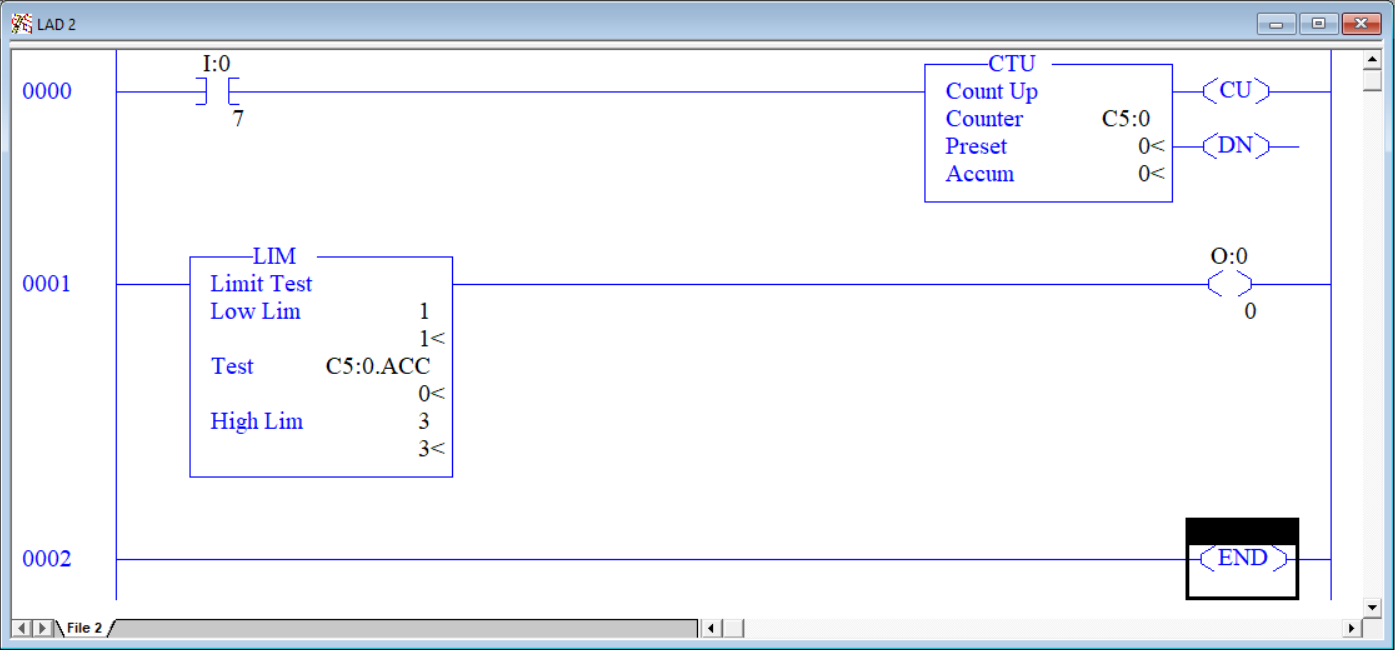


**Devices**

|  |  |  |
| --- | --- | --- |
| Inputs | | |
| *Device* | *Description* | *Symbol* |
| NO Pushbutton (PB2) | Counter | CNT |
| Outputs | | |
| *Device* | *Description* | *Symbol* |
| Green Pilot Light | LES Function Result | LIM\_LT |

**Instructions**

Modify the program as shown below.



1. After downloading the program, notice that with C5:0.ACC = 0, the green pilot light is off since 0 < 1.
2. Press PB2. With C5:0.ACC = 1 the green pilot light is on since 1 ≤ 1 ≤ 3.
3. Press PB2. With C5:0.ACC = 2 the green pilot light is on since 1 ≤ 2 ≤ 3.
4. Press PB2. With C5:0.ACC = 3 the green pilot light is on since 1 ≤ 3 ≤ 3.
5. Press PB2. With C5:0.ACC = 4 the green pilot light is off since 4 > 3.