

2.0 Counter Instruction

Use counter instruction (CTUD) to count activities as they occur, like products passing over a switch on a conveyor belt or pushes of a button. The counter instruction uses the counter data type to control the counter instruction.

The value in Current can count up past the preset value and down past zero. However, the value in Current cannot exceed the upper limit or go below the lower limit of a double integer.

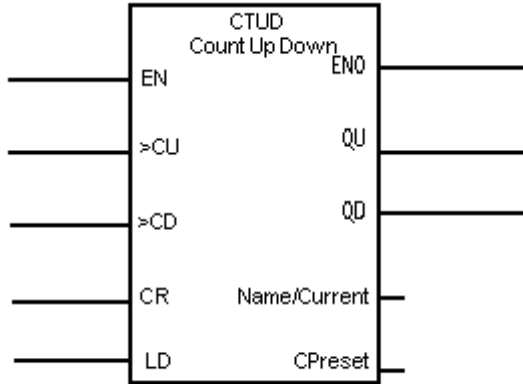
The QU and QD outputs do not change state unless the instruction is executed, even if the CPreSet has been modified to be less than or equal to the current value.

IMPORTANT

Global counters must be entered into the variable configurator as five-element, double-integer, non-volatile arrays.

Example: COUNTER1!(4).

2.1 Count Up Down (CTUD)



Use this instruction to increment or decrement a counter.

When EN is true, the instruction:

- increments the double integer value stored in Current for every false-true transition of the CU input
- decrements the double integer value stored in Current for every false-true transition of the CD input

The QD output becomes true when the value stored in Current is less than or equal to zero. The QU output becomes true when the value stored in Current is greater than or equal to the value stored in CPreset.

2.1.1 Input Parameters for the Count Up Down Instruction

This table lists the inputs for the CTUD instruction and the variable type and data type/range that each input supports.

Parameter	Description	Variable Type	Data Type/Range
EN	When this input is true, the instruction is enabled. When EN is false, the instruction is not executed and ENO, QU, and QD are set false.	Connect a Boolean input or output.	
CU	When the instruction is enabled and CR and LD are false, a false-true transition of this input causes the value in Current to be incremented by one. This input is false when not connected.		
CD	When the instruction is enabled and CR and LD are false, a false-true transition of this input decrements the value in Current by one. This input is false when not connected.		
CR	When this input is true, the value in Current is reset to zero. This input is false when it is not connected. See "Resetting a Counter," section 2.2.		
LD	When this input is true and CR is false, the value in Current is set equal to the value in CPreset. This input is false when not connected.		

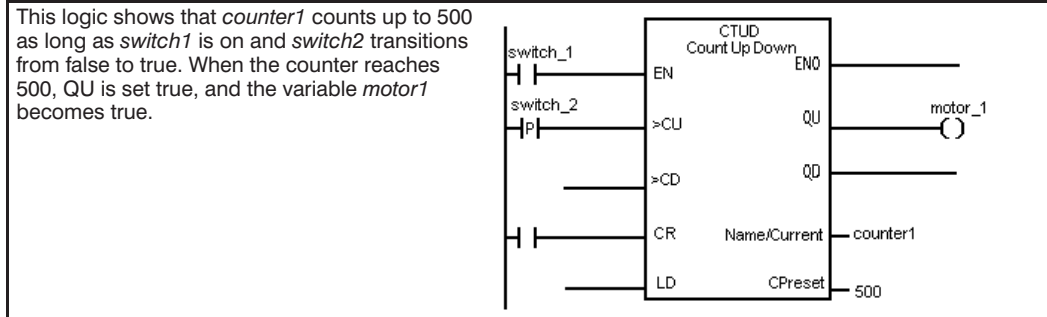
Parameter	Description	Variable Type	Data Type/Range
Name	Enter the name of the counter variable you want to use for this instruction.	data structure	counter See also "About Counter Variables"
CPreset	Enter the value you want the counter to count up to or count down from. This value is stored in <i>name.CPreset</i> . See "Changing the Preset Value of a Counter Instruction by Using Ladder Logic," section 2.3.	constant	double integer

2.1.2 Output Parameters for the Count Up Down Instruction

This table lists the outputs for the CTUD instruction. To use them, connect them to a contact, coil, or Boolean input of another instruction.

Parameter	Description
ENO	Use this output as the input to another instruction for easily chaining multiple instructions. This output follows the state of EN.
QU	This output is true when the value in Current is greater than or equal to the preset value (CPreset).
QD	This output is true when the value in Current is less than or equal to 0.

2.1.3 Example of a Count Up Down Instruction



2.2 Resetting a Counter

To reset a counter, attach a Boolean parameter to CR so that when the Boolean parameter is true, the value stored in Current is reset to 0. Any transitions to CU or CD are ignored.

2.3 Changing the Preset Value of a Counter Instruction by Using Ladder Logic

You can change the preset value of a counter instruction without having to edit the instruction in the AutoMax Ladder Editor. This is useful for frequently loading different preset values into a counter.

To change the preset value by using ladder logic

- Step 1. Place a Move Source Data to Destination (MOVE) instruction in the ladder program.
- Step 2. In the In input, enter the value you want to use as a new counter preset value.
- Step 3. In the Out output, enter the name of the counter's preset input (*name.CPreset*).
- Step 4. Condition the EN input of the MOVE instruction so that the new counter preset is loaded into the Counter instruction.

Tip

You can also use other instructions that have a double integer output to change a counter preset. For example, you can use an ADD instruction to calculate a new preset and place the result in counter's preset input (*name.CPreset*).