

## 3.0 Timer Instructions

Use the Timer instruction to enable and disable activities at pre-defined times. For example, set a timer to turn on a valve or shut off a furnace.

Choose from these timer instructions:

<b>Use this instruction:</b>	<b>To:</b>
Retentive Timer On (RTO)	track accumulated time
Timer Off Delay (TOF)	stop an activity at a preset time interval
Timer On Delay (TON)	start an activity at a preset time interval
Timer Pulse (TP)	enable an output for a preset time interval

The timer instructions use the timer data type for the variable in the Name parameter.

The maximum time interval is 248.5 days (5965 hours). You can specify time in increments of 0.01 seconds.

### How Timer Instructions Operate

Once the instruction is enabled, the elapsed value is updated at the start of a program's scan. For example, if a program is scheduled to run every second, the elapsed value for a timer in that program is incremented by 100 each time the program runs. Although the timer is specified in units of 0.01 seconds, the actual duration may be affected by the program's scan time. For example, if the scan time is 0.1 seconds and you set a timer to 0.01 seconds, the actual timer elapsed time will be 0.1seconds, because the timer output will only be updated when the timer executes at 0.1 seconds intervals (the program's scan time).

The T and Q outputs of the timer instructions do not change unless the timer instruction is executed, even if the correct amount of time has elapsed.

### **Guidelines for Programming Timer Instructions**

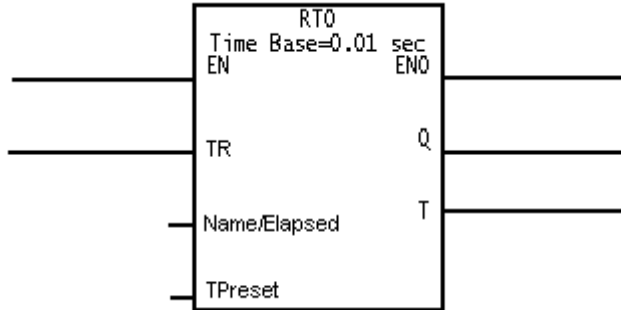
- When inserting timer instructions into a program, assign unique timer variables to each timer instruction. Do not use the same timer variable on more than one instruction in a program. However, you can use individual timer elements (name.Q, name. Elapsed, etc.) as variables on other contacts and instructions.
- When you remove a timer instruction while testing your edits on an online, active program, the Editor considers the timer instruction as being disabled. Therefore, should you ever re-instate that timer instruction, it is inserted into the program in a reset state. This means that the timer has lost its accumulated time.

#### **IMPORTANT**

Observe the following programming practices when creating ladder programs:

- Avoid using timer instructions in programs that are not executed at periodic intervals because their behavior will be unpredictable.
- Avoid skipping timer instructions using a JMP instruction because the timer's output will not be set unless the timer instruction is executed.
- Do not use timer instructions that use the same global, timer data structure in multiple programs because the timer will gain time.
- You must enter global timers into the variable configurator as five-element, double integer, non-volatile arrays. Example: TIMER1!(4).

### 3.1 Retentive Timer On (RTO)



Use this instruction to set an event at a preset interval. This instruction retains the Elapsed value after EN goes false and resumes keeping time when EN is true again. The timer can stop and start without the Elapsed value being reset.

When EN is true, the instruction begins incrementing the value in Elapsed. When the value in Elapsed equals the value in TPreset, output Q becomes true. Reset this instruction by setting input TR.

The value in Elapsed is in increments of 0.01 seconds.

### 3.1.1 Input Parameters for the RTO Instruction

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

Parameter	Description	Variable Type	Data Type/Range
EN	While this input is true, the instruction executes. When this input is false, the instruction is not executed and ENO is false.	Connect a Boolean input or output.	
TR	To reset the counter to 0, set this input to true. Output Q becomes false.		
Name	Enter the name of the timer variable you want to use for this timer.	data structure	timer See also "About Timer Variables"
TPreset	Enter the value that Elapsed must reach before the instruction sets output Q. Enter a value in increments of 0.01 seconds up to 248.5 days. This value is stored in <i>name</i> .TPreset.	constant	double integer (0 to 2147483647) 0 to 248.5 days in 0.01 second intervals

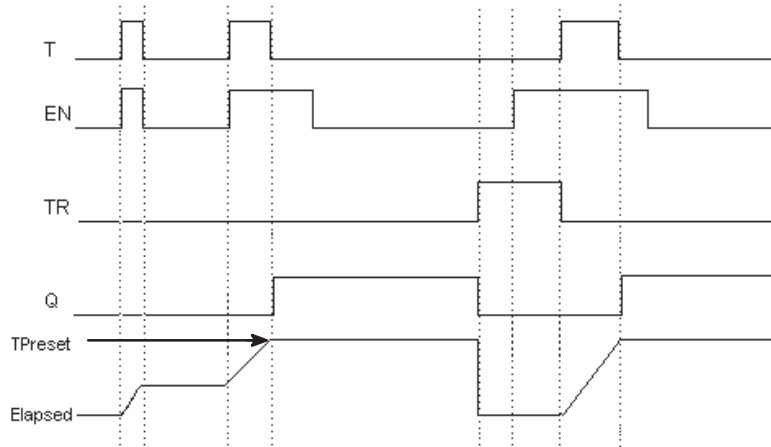
### 3.1.2 Output Parameters for the RTO Instruction

This table lists the outputs for the RTO instruction and the variable type and data type/range that each output supports.

Parameter	Description	Variable Type	Data Type/Range
ENO	Use this bit as the input to another instruction for easily chaining multiple instructions. This output follows the state of EN.	Connect a contact, coil, or Boolean input to another instruction.	
Q	This output is true when the value in Elapsed equals the value in TPreset. Q is false when R is true.	Connect a Boolean input or a coil.	
T	This output is true when all of the following conditions are met: <ul style="list-style-type: none"> <li>the instruction is enabled</li> <li>the value in Elapsed is less than the value in TPreset</li> <li>TR is false</li> </ul> T is false when the instruction is disabled or the value in Elapsed equals the value in TPreset.		
Elapsed	This value is specified in increments of 0.01 seconds. This value is incremented when EN is true and until its value equals that of TPreset. The value in Elapsed is reset only when TR is true.  The value in Elapsed is stored in the element <i>name</i> .Elapsed and will not exceed the TPreset value when the instruction is enabled.	constant	double integer (0 to 2147483647) 0 to 248.5 days in 0.01s intervals

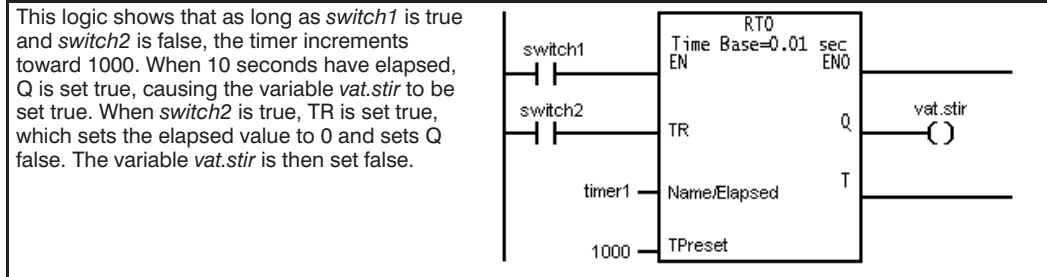
### 3.1.3 RTO Timing Diagram

The following diagram shows the interaction between the RTO instruction inputs and outputs at various time intervals.

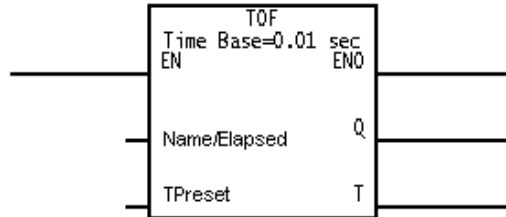


RTO Timing Diagram

### 3.1.4 Example of an RTO Instruction



### 3.2 Timer Off Delay (TOF)



Use this instruction to disable an activity at a preset interval. When EN is false, the instruction increments the value in Elapsed until it reaches the value you defined in TPreset. When the value in Elapsed equals the value in TPreset, output Q goes false. The value in Elapsed is in increments of 0.01 seconds.

### 3.2.1 Input Parameters for the TOF Instruction

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

Parameter	Description	Variable Type	Data Type/Range
EN	While this input is false, the instruction begins the timer operation. When EN is true, the value in Elapsed is reset to zero.	Connect a Boolean input or output.	
Name	Enter the name of the timer variable you want to use for this timer.	data structure	timer See also "About Timer Variables"
TPreset	Enter the value that Elapsed must reach before the instruction resets output Q. Enter a value in increments of 0.01 seconds up to 248.5 days. This value is stored in <i>name</i> .TPreset.	constant	double integer (0 to 2147483647) 0 to 248.5 days in 0.01 second intervals

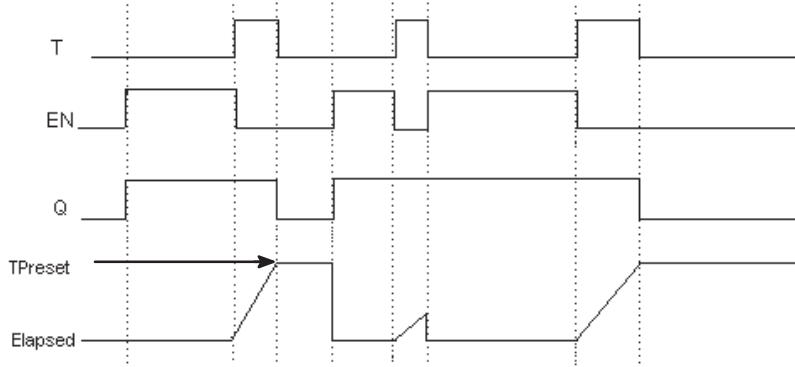
### 3.2.2 Output Parameters for the TOF Instruction

This table lists the outputs for the TOF instruction and the variable type and data type/range that each output supports.

Parameter	Description	Variable Type	Data Type/Range
ENO	Use this output as the input to another instruction for easily chaining multiple instructions. ENO follows the state of EN.	Connect a contact, coil, or Boolean input to another instruction.	
Q	This output is false when the value in Elapsed equals the value in TPreset. This output is true while EN is true and Elapsed is less than TPreset.		
T	This output is true while EN is false and the value in Elapsed is less than the value in TPreset. T is false while EN is true or the value in Elapsed equals the value in TPreset.		
Elapsed	This value is specified in increments of 0.01 seconds. This value is incremented when EN is false, until its value equals that of TPreset. The value in Elapsed is reset to zero when EN is true. The value in Elapsed is stored in the element <i>name</i> .Elapsed and will not exceed the preset value when the instruction is enabled.	constant	double integer (0 to 2147483647) 0 to 248.5 days in 0.01 second intervals

### 3.2.3 TOF Timing Diagram

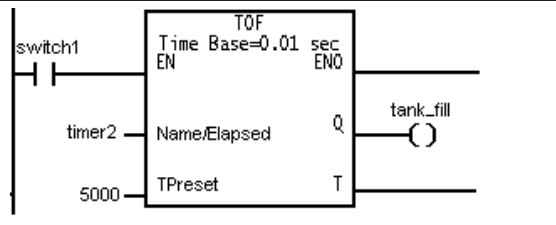
The following diagram shows the interaction between the TOF instruction inputs and outputs at various time intervals.



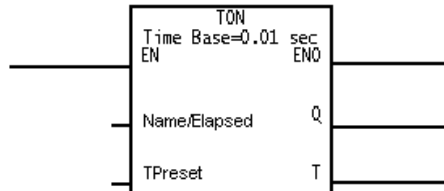
TOF Timing Diagram

### 3.2.4 Example of a TOF Instruction

This logic shows that as long as *switch1* is false, *timer2* increments to 5000. When 50 seconds have elapsed, the timer sets *Q* false, causing the variable *tank\_fill* to be set false.



### 3.3 Timer On Delay (TON)



Use this instruction to enable an activity at a preset interval. While EN is true, the instruction increments the value in Elapsed until it reaches the value you defined in TPreset. When the value in Elapsed equals the value in TPreset, output Q is set.

The value in Elapsed is in increments of 0.01 seconds.

### 3.3.1 Input Parameters for the TON Instruction

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

Parameter	Description	Variable Type	Data Type/Range
EN	While this input is true, the instruction begins the timer operation. When EN is false, the value in Elapsed is reset to zero.	Connect a Boolean input or output.	
Name	Enter the name of the timer variable you want to use for this timer.	data structure	timer See also "About Timer Variables"
TPreset	Enter the value that Elapsed must reach before the instruction sets the output Q. Enter a value in increments of 0.01 seconds up to 248.5 days. This value is stored in <i>name</i> .TPreset.	constant	double integer (0 to 2147483647) 0 to 248.5 days in 0.01 second intervals

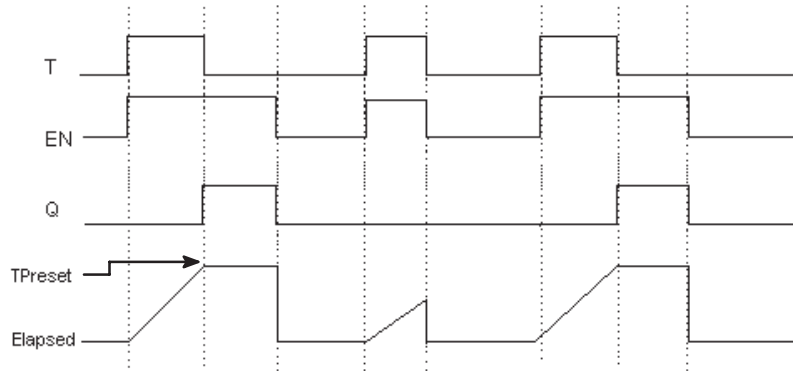
### 3.3.2 Output Parameters for the TON Instruction

This table lists the outputs for the TON instruction and the variable type and data type/range that each output supports.

Parameter	Description	Variable Type	Data Type/Range
ENO	Use this output as the input to another instruction for easily chaining multiple instructions. ENO follows the state of EN.	Connect a contact, coil, or Boolean input to another instruction.	
Q	This output is true when the instruction is enabled and Elapsed is equal to TPreset. Q is false when EN is false or when Elapsed is less than TPreset.		
T	This output is true while EN is true and the value in Elapsed is less than the value in TPreset. T is false when EN is false or the value in Elapsed is equal to the value in TPreset.		
Elapsed	This value is specified in increments of 0.01 seconds. This value is incremented when EN is true, until its value equals that of TPreset. The Elapsed value is reset to zero when EN is false.  The value in Elapsed is stored in the element <i>name</i> .Elapsed and will not exceed the TPreset value when the instruction is enabled.	constant	double integer (0 to 2147483647) 0 to 248.5 days in 0.01 second intervals

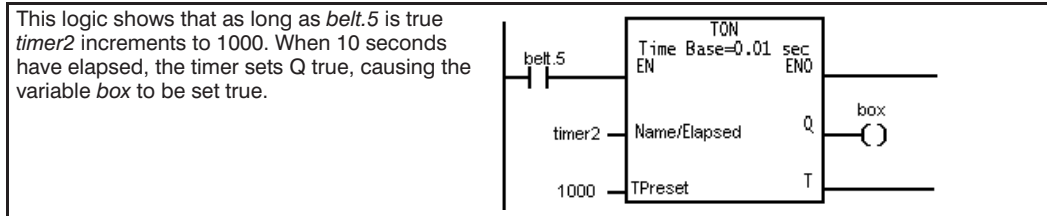
### 3.3.3 TON Timing Diagram

The following diagram shows the interaction between the TON instruction inputs and outputs at various time intervals.

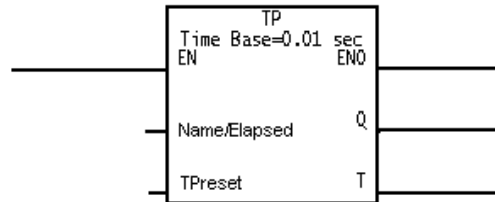


**TON Timing Diagram**

### 3.3.4 Example of a TON Instruction



### 3.4 Timer Pulse (TP)



Use this instruction to enable an output for a preset amount of time. This instruction guarantees that an output remains on for a preset time, regardless of the state of the EN input.

When EN is true, the instruction sets outputs Q and T and begins counting towards the preset value. The timer increments regardless of EN's state until the value in Elapsed equals that of TPreset. When these values are equal, outputs Q and T become false. The value in Elapsed is reset when EN is false and the value in Elapsed equals that of TPreset. The value in Elapsed is in increments of 0.01 seconds.

### 3.4.1 Input Parameters for the TP Instruction

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

Parameter	Description	Variable Type	Data Type/Range
EN	While this input is true, the instruction begins the timer operation. When EN is false and TPreset is equal to Elapsed, the value in Elapsed is reset to zero.	Connect a Boolean input or output.	
Name	Enter the name of the timer variable you want to use for this timer.	data structure	timer See also "About Timer Variables"
TPreset	Enter the value that Elapsed must reach before the instruction resets outputs Q and T. Enter a value in increments of 0.01 seconds up to 248.5 days. This value is stored in <i>name</i> .TPreset.	constant	double integer (0 to 2147483647) 0 to 248.5 days in 0.01 second intervals

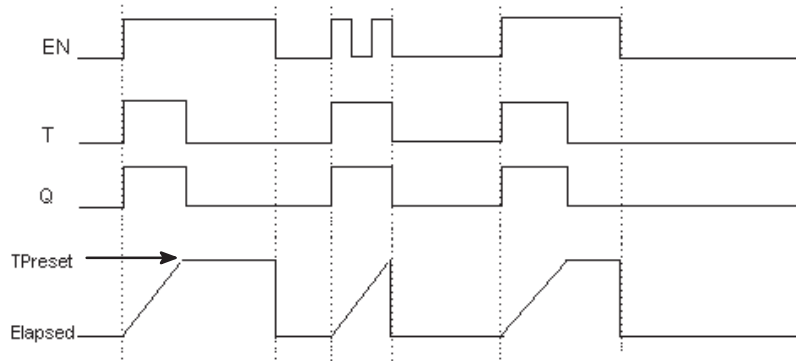
### 3.4.2 Output Parameters for the TP Instruction

This table lists the outputs for the TP instruction and the variable type and data type/range that each output supports.

Parameter	Description	Variable Type	Data Type/Range
ENO	Use this output as the input to another instruction for easily chaining multiple instructions. ENO follows the state of EN.	Connect a contact, coil, or Boolean input to another instruction.	
Q, T	These outputs are set true when EN is true and the value in Elapsed is less than that in TPreset. Q and T remain true while the value in Elapsed is less than the value in TPreset.  These outputs are set false when the value in Elapsed equals the value in TPreset.		
Elapsed	This value is specified in increments of 0.01 seconds. This value is incremented when EN is true, until its value equals that of TPreset. The Elapsed value is reset to zero when EN is false and Elapsed and TPreset are equal.  The value in Elapsed is stored in the element <i>name</i> . Elapsed and will not exceed the TPreset value when the instruction is enabled.	constant	double integer (0 to 2147483647)  0 to 248.5 days in 0.01 second intervals

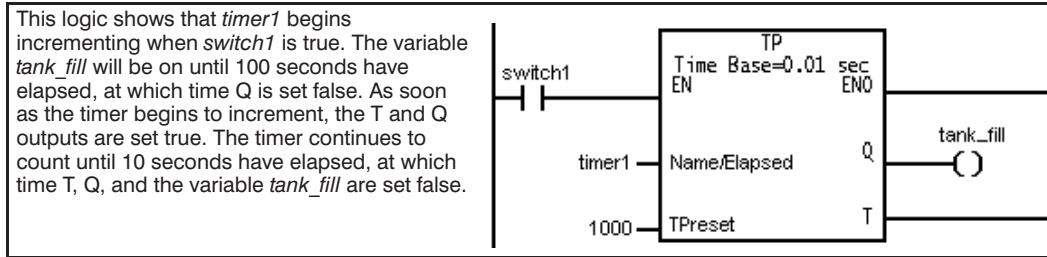
### 3.4.3 TP Timing Diagram

The following diagram shows the interaction between the TP instruction inputs and outputs at various time intervals.



TP Timing Diagram

### 3.4.4 Example of a TP Instruction



### 3.5 Changing the Preset Value of a Timer Instruction by Using Ladder Logic

You can change the preset value of a timer instruction without having to edit the instruction in the AutoMax Ladder Editor. This is useful for frequently loading different preset values into a timer. You can change the preset value of a timer by using a ladder logic instruction with a double integer output for global and local timer presets.

### 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 timer preset value.
- Step 3. In the Out output, enter the name of the timer's preset input (*name.TPreset*).
- Step 4. Condition the EN input of the MOVE instruction so that the new timer preset is loaded into the timer instruction.

#### Tip

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