

Introduction to Programmable Logic Controllers – Part I

Module 4: Timers and Counters

In this module, PLC timer and counter instructions are discussed. After studying this module, the student should be able to:

- List the types of timers and counters used in ladder logic programs;
- Understand how the timers and counters work in ladder logic programs;
- Use status bits of timers and counters to control other instructions;
- Write a ladder logic program using timers and counters.

Three types of timers are used in PLC ladder logic programs. They are ON-delay timers, OFF-delay timers, and retentive timers. Figures 4.1, 4.2, and 4.3 show the timer instructions used in the Allen-Bradley PLC.

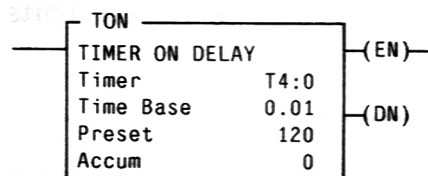


Figure 4.1 ON-Delay Timer Instruction

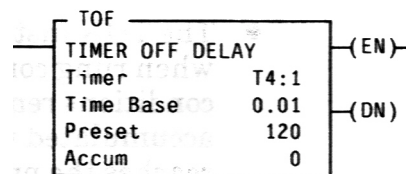


Figure 4.2 OFF-Delay Timer Instruction

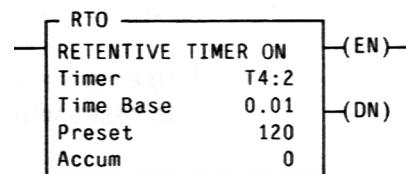


Figure 4.3 Retentive Timer Instruction

When programming a timer instruction, the programmer must specify the Timer address, the Time Base, and the Preset value, which are listed in the instruction. The format of the Timer address is T4:N, where N is a positive integer. Each timer instruction should have a unique number that distinguishes its timer instruction from other timer instructions. The Time Base value is an interval that the timer is going to use. This value can be set to 1 second, 0.01 second, or 0.001 second. The Preset value specifies how many intervals a timer should count before the

timing is complete, also known as “done”. A timer’s setting time equals its Preset value multiplied by its Time Base. For example, if a timer’s Time Base is 0.01 and Preset is 500, the timer’s setting time is $500 \times 0.01 \text{ second} = 5 \text{ seconds}$. That means this timer will be done 5 seconds after the timer instruction is enabled.

A timer instruction must be located next to the right rail in a rung. An ON-delay timer is enabled when its rung is true. A rung is true when there is at least one path made by the instructions that are true from the left rail to the timer instruction. An OFF-delay timer is enabled when its rung is false. When a timer is enabled, its Accumulate value shows how many Time Base intervals have elapsed since the timer was activated. A timer is *done* when its Accumulate value reaches its Preset value. When an ON-delay timer or an OFF-delay timer is timing, its rung condition change, i.e. a rung changes from true to false for an ON-delay timer or a rung changes from false to true for an OFF-delay timer, will cause the timer to stop and its Accumulate value to be reset to zero.

A retentive timer works like an ON-delay timer with one difference. That is, when its rung condition changes from true to false, the timer simply stops timing, but its Accumulate value is not reset to zero. When its rung condition goes from false to true again, the retentive timer’s Accumulate value counts up from where it stopped the last time. To reset a retentive timer’s Accumulate value to zero, a *reset* instruction with the same timer address must be used. A reset instruction is a controlled instruction, which means it must be located next to the right rail in a rung. When its rung is true, the reset instruction becomes enabled and resets the timer according to the address that is assigned to the reset instruction.

Each timer instruction has three very useful status bits. These bits are Timer Enable (TE), Timer Timing (TT) and Timer Done (TN). Each of these bits has one bit of memory and the memory is affected by the corresponding bit status. For an ON-delay timer and a retentive timer,

- The Timer Enable bit is high when the timer’s rung is true; it is low when the rung is false.
- The Timer Timing bit is high when the timer’s rung is true and the Accumulate value is less than the Preset value. This bit is low when the rung is false or after the Accumulate value equals the Preset value.
- The Timer Done bit is high when the rung is true and the timer is done. It is low when the rung is false or before the timer is done.

For an OFF-delay timer,

- The Timer Enable bit is high when the timer’s rung is false; it is low when the rung is true.
- The Timer Timing bit is high when the timer’s rung is false and the Accumulate value is less than the Preset value. This bit is low when the rung is true or after the Accumulate value equals the Preset value.

- The Timer Done bit is high when the rung is false and the timer is done. It is low when the rung is true or before the timer is done.

Figure 4.4 is a timing diagram of an ON-delay timer's control bits. In this diagram, the timer is disabled after its Accumulate value reaches its Preset value.

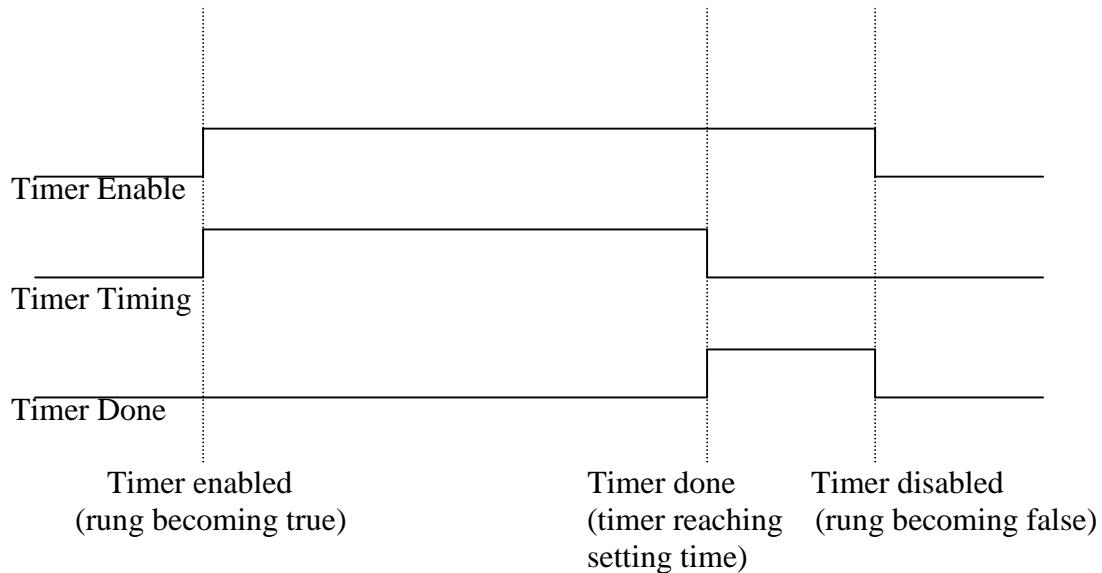


Figure 4.4 An Example of a Timer's Control Bit Logic

Figure 4.5 is another timing diagram of an ON-delay timer. This timer is disabled before its Accumulate value reaches its Preset value. The Timer Done bit in this diagram is always low because the timer never reaches a done status.

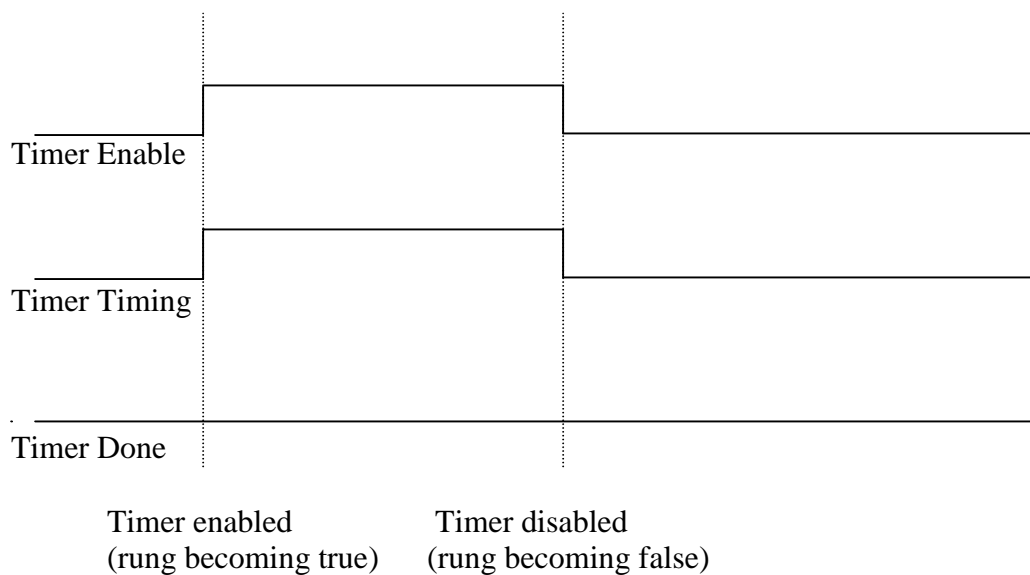


Figure 4.5 Another Example of a Timer's Control Bit Logic

Figure 4.6 is a PLC ladder logic program that uses timer instructions to alternately turn-on two output devices. A start button is connected to the input terminal I:2/1 and a stop button is connected to the input terminal I:2/2. When the start button is pushed, it energizes the instruction B3:1/1 in the first rung. This instruction latches itself by using the Exam-if-Closed instruction B3:1/1 in the same rung. That allows the push button to be released and the B3:1/1 stays in HIGH status. When B3:1/1 is HIGH, it enables the timer instruction T4:1 in the second rung. Timer T4:1 is preset to 2 minutes (120 seconds). When this timer is timing, the T4:1/TT instruction in rung 4 is true. It energizes the output O:3/4. When this timer reaches 2 minutes, T4:1/TT goes to false and de-energizes O:3/4. However, the T4:1/DN bit goes to HIGH at this moment. That enables the timer T4:2 that is preset to 3 minutes. When T4:2 is timing, the T4:2/TT instruction in rung 5 turns on the output O:3/5. When timer T4:2 reaches 3 minutes, T4:2/TT goes to LOW and O:3/5 is de-energized. Meantime, the T4:2/DN bit goes to HIGH. That causes the second rung's condition to be false and the timer T4:1 to be reset. When timer T4:1 is reset, the T4:1/DN bit goes to low and timer T4:2 in the third rung is reset. This causes T4:2/DN to be low. In the next scan cycle, the second rung turns to true and the timer T4:1 is timing again. This starts another cycle of energizing the outputs. These two outputs alternate between ON and OFF until the stop button is pushed. Pushing the stop button de-energizes B3:1/1 in the first rung. That breaks the latch in the first rung. It also stops and resets the timers T4:1 and T4:2.

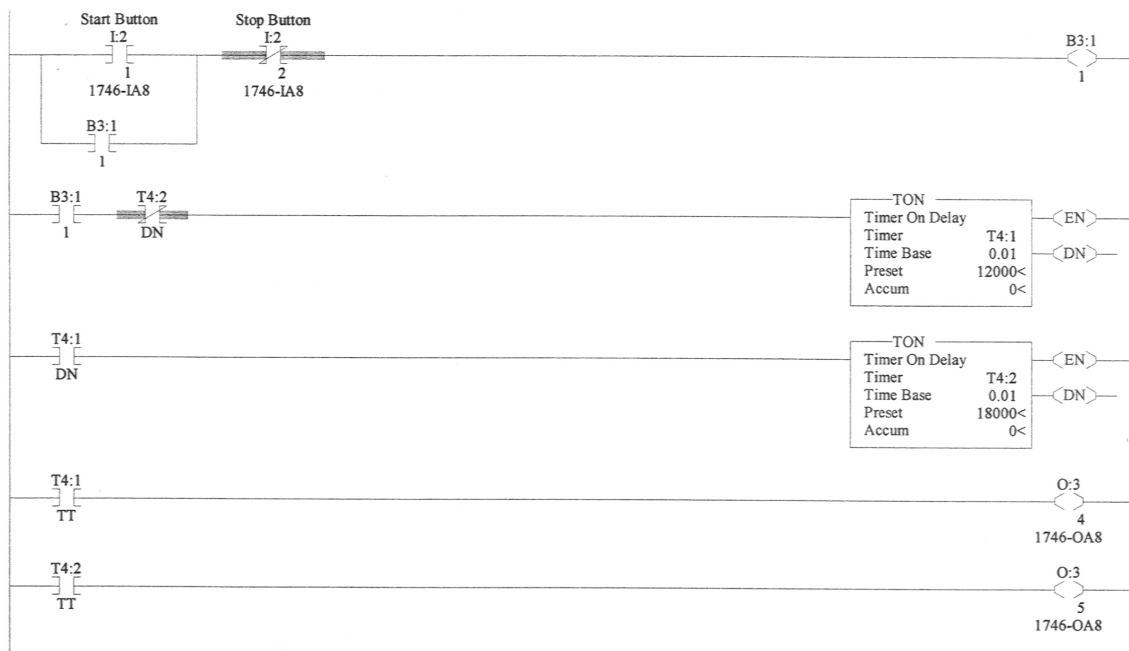


Figure 4.6 A Program Using Timers to Control Outputs

A PLC counter instruction can be a count-up instruction or a count-down instruction. These are shown in Figure 4.7 and Figure 4.8. When a counter instruction is used in a program, the programmer must specify the counter address. A counter address has the format of C5:N, where N is a positive integer to distinguish it from other counters. The programmer must also specify the Preset value, which is a signed (positive or negative) integer.

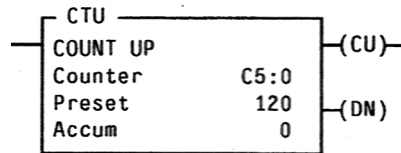


Figure 4.7 Count-up Counter Instruction

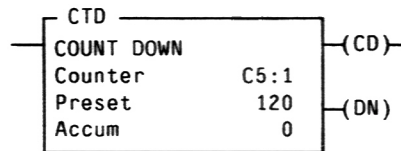


Figure 4.8 Count-down Counter Instruction

A counter instruction counts false-to-true rung transitions. These rung transitions could be caused by events monitored by the program or by something else. Each time a rung condition is changing from false to true, a count-up instruction in the rung increases its Accumulate value by 1 and a count-down instruction in the rung decreases its Accumulate value by 1.

There are several status bits associated with each counter instruction. Three of these bits are used frequently in PLC ladder logic programs. These bits are the counter-up enable bit (CU), the counter-down enable bit (CD), and the done bit (DN).

- When a count-up counter is counting, its CU bit is high.
- When a count-down counter is counting, its CD bit is high.
- A DN bit is high when its counter's Accumulate value is greater or equal to the counter's Preset value.

A counter retains its Accumulate value when its rung condition is true, false, or changing from true to false. A counter's Accumulate value can be reset to zero by using a reset instruction that has the same counter address. The counter is reset when the reset instruction is enabled.

The program in Figure 4.9 is used for counting parts and controlling box packaging in a manufacturing company. One hundred parts are packed into a box. A sensor, which is connected to input terminal I:2/1, detects the parts coming down from a conveyer. When a part passes in front of it, the sensor has a high output. This high output causes counter C5:1 to count. When this counter is counting up to 100, its DN bit goes to high. That causes the output O:3/1 in rung 2 to be energized, which initiates a box packaging process that is not included in this example program. The switching of DN bit of counter C5:1 to high also causes the counter C5:2 in rung 3 to count. This counter counts the number of boxes packed. When C5:2 is counting, its CU bit is high. That enables the reset instruction in rung 4 to reset counter C5:1's Accumulate value to 0. The C5:1 will count up from 0 again.

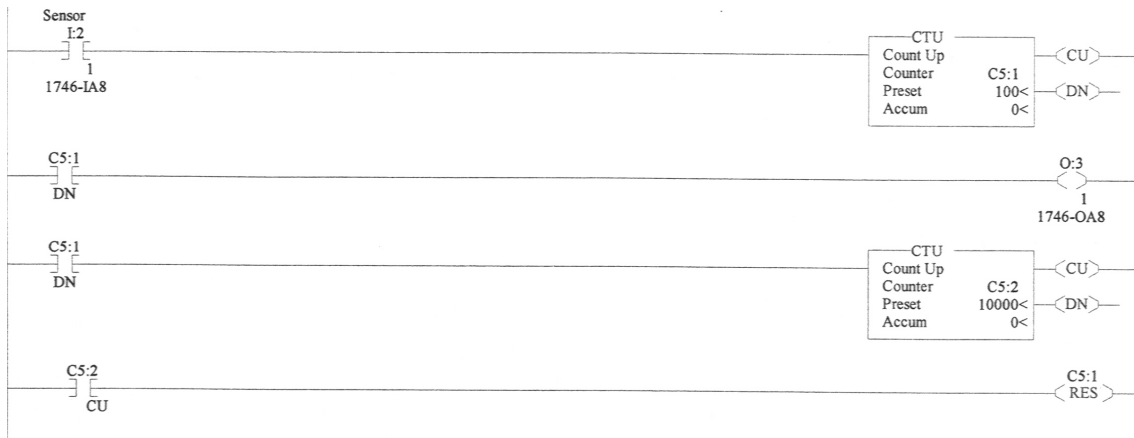


Figure 4.9 A Program Using Counters to Count Parts

Related web sites:

<http://www.plcs.net>

<http://www.sea.siemens.com/step/templates/lesson.mason?plcs:1:1:1>