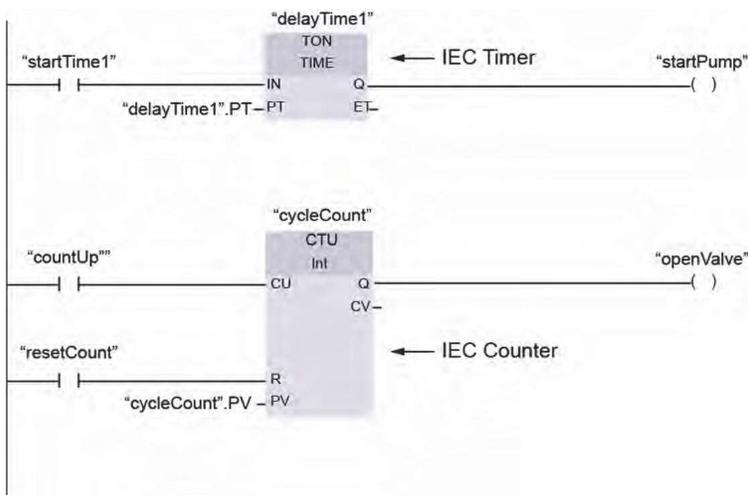


## Timers and Counters

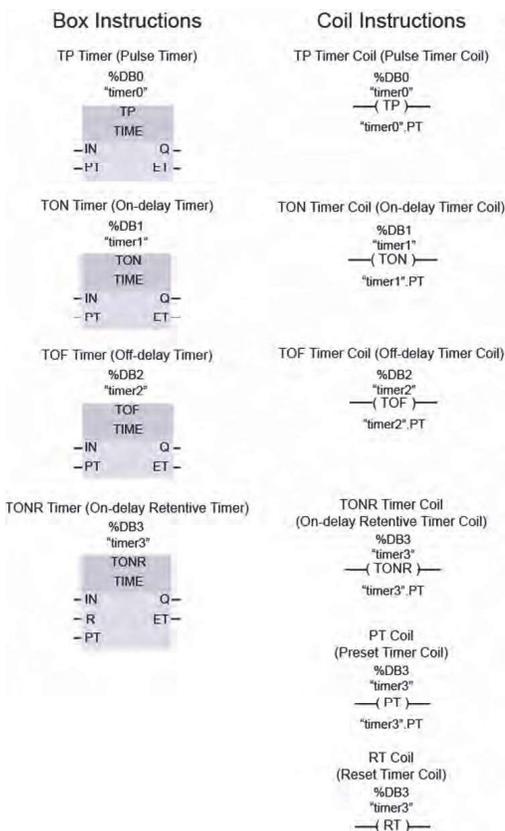


Timers and counters are extremely useful PLC instructions. Timers allow PLCs to control and monitor time-dependent aspects of machines and processes. Counters allow PLCs to keep track of actions or events.

The IEC 61131 standard identifies PLC programming languages and standardizes the instructions available. Some Siemens S7 PLCs were already in use when the IEC 61131 specification was developed. This is noticeable when reviewing the SIMATIC timer and SIMATIC counter instructions available for S7-300 and S7-400 PLCs. While the tasks performed by SIMATIC timers and counters are similar to those performed by IEC timers and counters, there are noticeable differences. Keep in mind that S7-300 and S7-400 PLCs also allow IEC timers and IEC counters to be implemented through use of system function blocks.

Because S7-1200 and S7-1500 PLCs were developed well after the deployment of IEC 61131, these PLCs include IEC timers and counters as basic instructions. Therefore, to minimize the complexity of this lesson, the IEC timers and counters included are consistent with those used with S7-1200 PLCs.

# IEC Timers



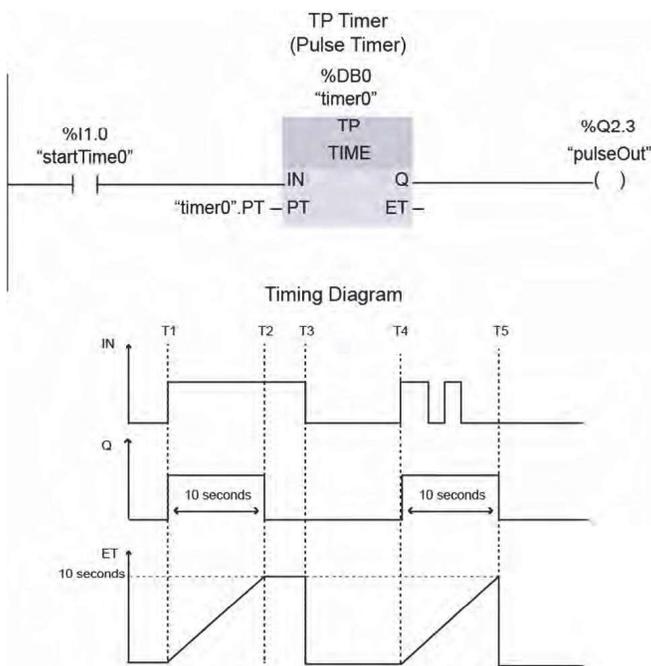
For LAD programs, S7-1200 PLCs support the IEC box and coil timer instructions shown in the accompanying graphic.

Each timer uses a data block (DB) to store timer data. For S7-1200 PLCs, the number of timers that you can use in program is limited only by CPU memory size.

IEC timers in S7-1200 PLCs use the 32-bit Time data type for preset time (PT) and elapsed time (ET) values. Time data is stored as a signed double integer interpreted as milliseconds. However, negative time values cannot be used with timer instructions. This means that the range of values that can be represented by timer preset and elapsed times is 0 to +2,147,483,647 milliseconds, which is equal to 24 days, 20 hours, 31 minutes, 23 seconds, and 647 milliseconds.

In STEP 7, preset time is entered using this format: T#(days)d(hours)h(minutes)m(seconds)s(milliseconds)ms. For example, T#12d10h30m21s253ms is a valid time, but it is not necessary to specify all units of time. Therefore, T#5h10s and T#500h are also valid.

# TP Timer Box Instruction

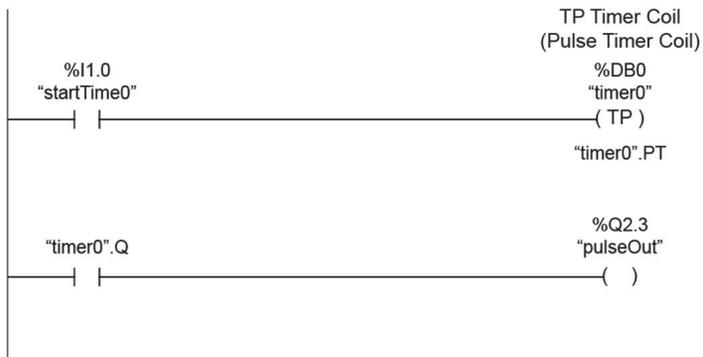


A TP timer, also called a pulse timer, is an instruction that generates a pulse at output parameter (Q) when the input parameter (IN) transitions from 0 to 1, unless the timer has already started timing. The pulse duration is defined by the preset time parameter (PT). The elapsed time (ET) increases from 0 to the preset time during the pulse duration and is retained until IN transitions from 1 to 0. The PT and ET values are stored in a data block (DB).

The accompanying graphic shows a TP timer in a network and an associated timing diagram. Note that PT is set to 10 seconds. When the I1.0 contact closes at time T1, IN transitions from 0 to 1. This starts the timer timing and causes Q and output Q2.3 to transition from 0 to 1. At time T2, when ET reaches 10 seconds, the timer stops timing, and Q and output Q2.3 are reset to 0. The 10 seconds elapsed time is retained until time T3 when the I1.0 contact opens, and IN and ET are reset to 0.

This sequence begins again at time T4 when the I1.0 contact closes. However, the TP timer continues to time and Q, and output Q2.3 remain on when IN is reset to 0 during the preset time. At T5, the elapsed time reaches 10 seconds and Q and output Q2.3 are reset to 0. Because IN is already at 0, ET is also reset to 0.

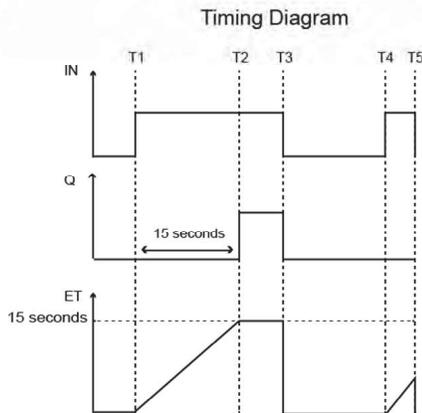
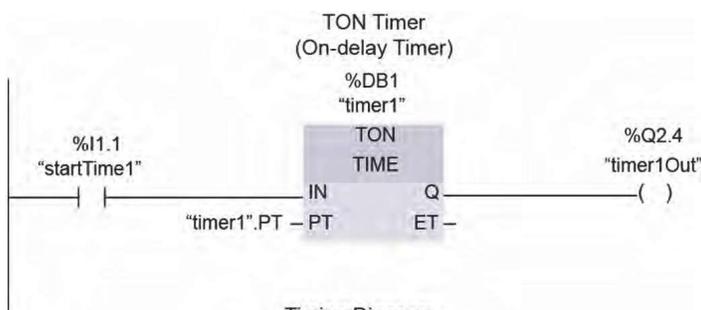
## TP Timer Coil Instruction



As shown in the accompanying graphic, a LAD program can also use a TP timer coil instruction to perform the same function as a TP timer box instruction.

Like the box instruction, a TP timer coil has an associated data block that stores the PT and ET values.

## TON Timer Box Instruction

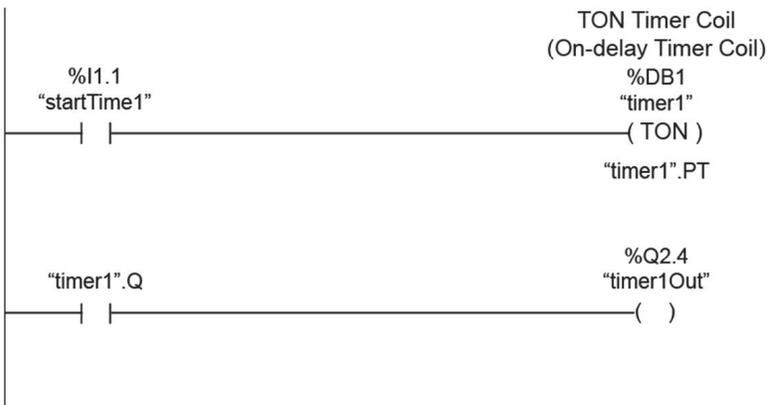


A TON timer, also called an on-delay timer, is an instruction that begins timing when the input parameter (IN) transitions from 0 to 1. When the elapsed time (ET) reaches the preset time (PT), the output parameter (Q) transitions from 0 to 1. Q remains a 1 until IN transitions from 1 to 0. If IN transitions from 1 to 0 before ET is equal to PT, the timer stops timing, ET is reset to 0, and Q remains a 0. The PT and ET values are stored in the data block (DB) associated with the instruction.

The accompanying graphic shows a TON timer in a network and an associated timing diagram. In this example, PT is equal to 15 seconds. At time T1, when the I1.1 contact closes, IN transitions from 0 to 1. This starts the timer timing. At time T2, when ET is equal to 15 seconds, the timer stops timing, and Q and output Q2.4 transition from 0 to 1. At time T3, when the I1.1 contact opens, IN, ET, Q, and output Q2.4 are reset to 0.

This sequence begins again at time T4 when the I1.1 contact closes and IN transitions from 0 to 1. However, at time T5, the I1.1 contact opens, and IN transitions from 1 to 0 before ET is equal to PT, so Q and output Q2.4 remain off, and ET is reset to 0.

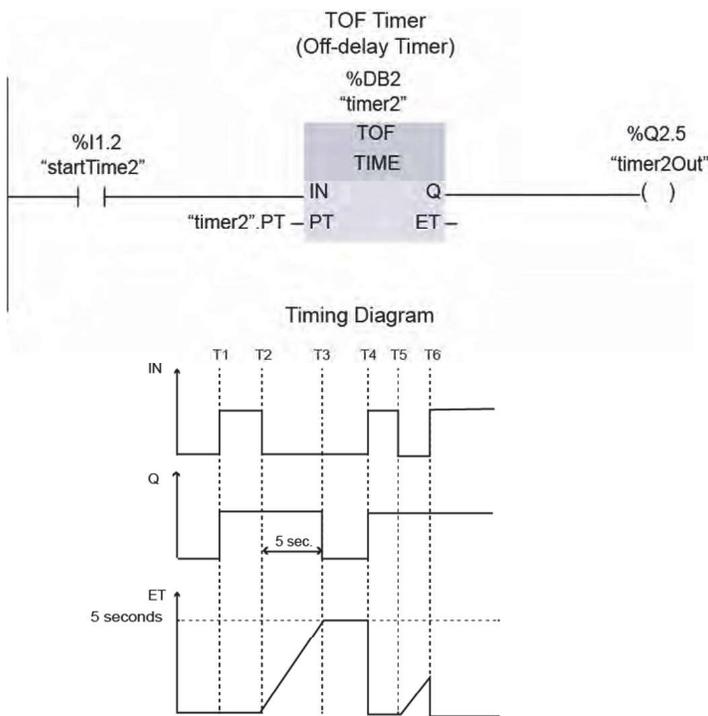
## TON Timer Coil Instruction



As shown in the accompanying graphic, a LAD program can also use a TON timer coil instruction to perform the same function as a TON timer box instruction.

Like the box instruction, a TON timer coil has an associated data block that stores the PT and ET values.

# TOF Timer Box Instruction

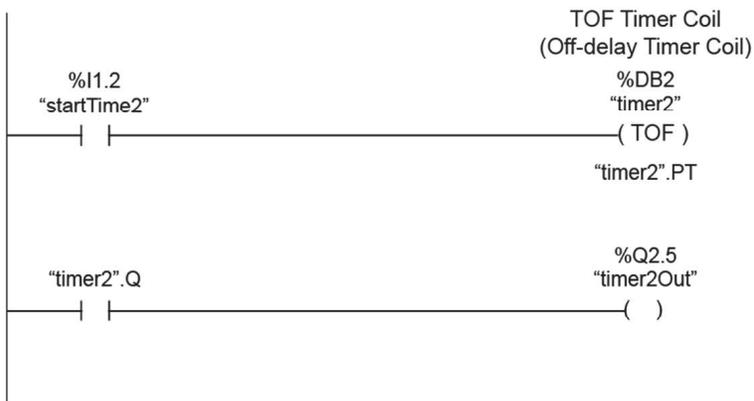


A TOF timer, also called an off-delay timer, is an instruction that sets the output parameter (Q) to 1 when input parameter (IN) transitions from 0 to 1. However, the TOF timer does not begin timing until IN transitions from 1 to 0. When the elapsed time (ET) reaches the preset time (PT), Q transitions from 1 to 0. Q remains a 0 and the elapsed time is retained until the next 0 to 1 transition of IN. The PT and ET values are stored in the data block (DB) associated with the instruction.

The accompanying graphic shows a TOF timer in a network and an associated timing diagram. In this example PT is equal to 5 seconds. When the I1.2 contact closes at time T1, IN, Q, and output Q2.5 transition from 0 to 1. At time T2, when the I1.2 contact opens, IN transitions from 1 to 0, and the timer starts timing. At time T3, when ET is equal to 5 seconds, the timer stops timing, and Q and output Q2.5 transition from 1 to 0.

At time T4, when the I1.2 contact closes and IN and Q transition from 0 to 1, ET is reset to 0. At time T5, when the I1.2 contact opens, IN transitions from 1 to 0, and the timer starts timing. However, at time T6, the I1.2 contact closes and IN transitions from 0 to 1 before ET is equal to PT, so Q and output Q2.5 remain on, and ET is reset to 0.

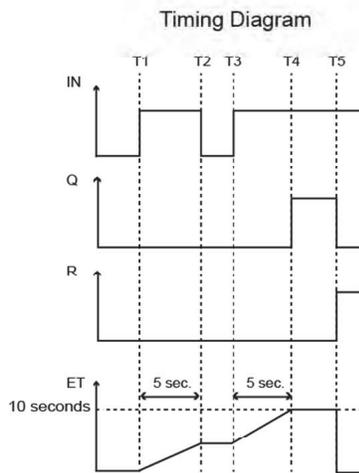
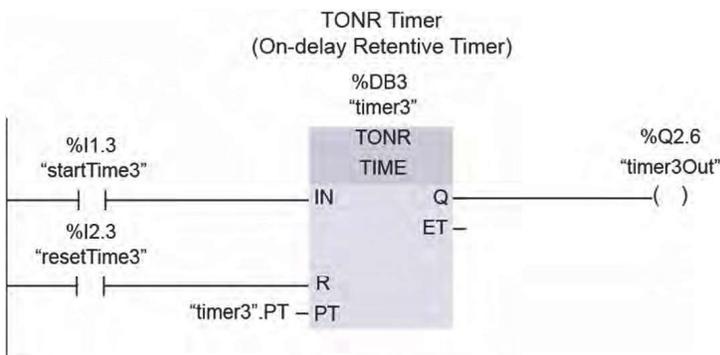
## TOF Timer Coil Instruction



As shown in the accompanying graphic, a LAD program can also use a TOF timer coil instruction to perform the same function as a TOF timer box instruction.

Like the box instruction, a TOF timer coil has an associated data block that stores the PT and ET values.

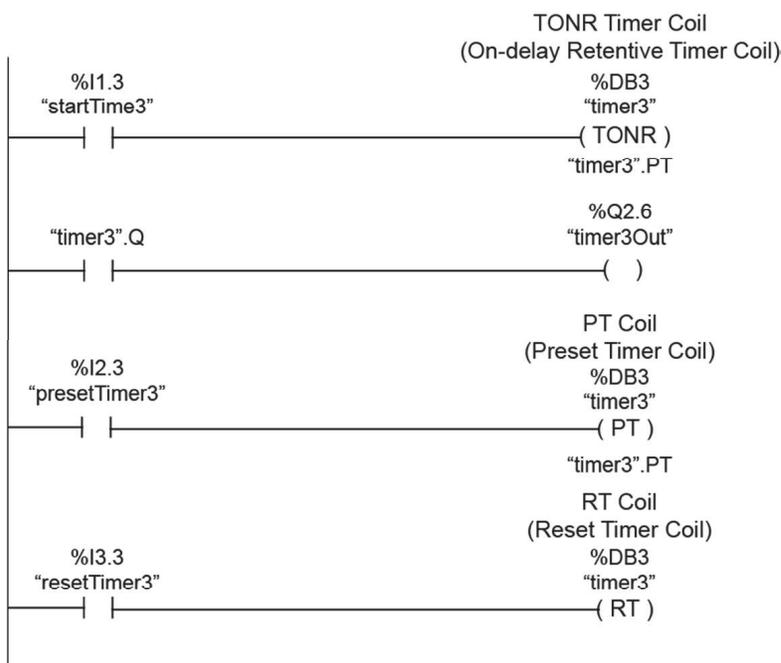
## TONR Timer Box Instruction



A TONR timer, also called an on-delay retentive timer, is an instruction that begins timing when the input parameter (IN) transitions from 0 to 1. If IN transitions from 1 to 0, the elapsed time (ET) is retained. When IN again transitions from 0 to 1, the timer resumes timing from the retained ET value. When ET reaches the preset time (PT), the timer stops timing, and the output parameter (Q) transitions from 0 to 1. When the reset parameter (R) transitions from 0 to 1, ET and Q are reset to 0, even if IN remains a 1. The PT and ET values are stored in the data block (DB) associated with the instruction.

The accompanying graphic shows a TONR timer in a network and an associated timing diagram. In this example PT is set to 10 seconds. When the I1.3 contact closes at time T1, IN transitions from 0 to 1. This starts the timer timing. At time T2, when the I1.3 contact opens, the timer stops timing, but retains the 5 second ET value. At time T3, when the I1.3 contact closes, IN transitions from 0 to 1, and the timer resumes timing. At time T4, when ET is equal to 10 seconds, the timer stops timing, and Q and output Q2.6 transition from 0 to 1. At time T5, when the I2.3 contact closes, R transitions from 0 to 1, and ET, Q, and output Q2.6 are reset to 0.

## TONR, PT, and RT Timer Coil Instructions



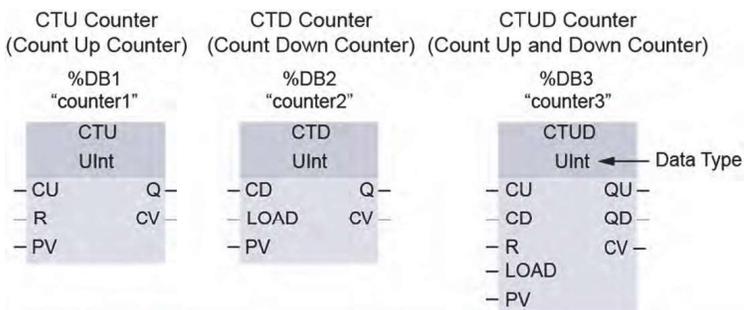
As shown in the accompanying graphic, a LAD program can also use a TONR timer coil instruction to perform the same function as a TONR timer box instruction. Like the box instruction, a TONR timer coil has an associated data block that stores the PT and ET values.

The accompanying graphic also shows a PT timer coil instruction, also called a preset timer coil instruction, and an RT timer coil instruction, also called a reset timer coil instruction.

These instructions can be used with timer box instructions or other timer coil instructions and can be placed in a mid-line or end position in a network.

When the PT timer coil instruction receives power flow, it loads a preset time to the specified timer. When the RT timer coil instruction receives power flow, it resets the specified timer.

## IEC Counters



| Integer Type                    | Size                    | Equivalent Decimal Range         |
|---------------------------------|-------------------------|----------------------------------|
| Unsigned Short Integer (USInt)  | 8 Bits (1 Byte)         | 0 to 255                         |
| Unsigned Integer (UInt)         | 16 Bits (1 Word)        | 0 to 65,535                      |
| Unsigned Double Integer (UDInt) | 32 Bits (1 Double Word) | 0 to 4,294,967,295               |
| Signed Short Integer (SInt)     | 8 Bits (1 Byte)         | -128 to +127                     |
| Signed Integer (Int)            | 16 Bits (1 Word)        | -32,768 to +32,767               |
| Signed Double Integer (DInt)    | 32 Bits (1 Double Word) | -2,147,483,648 to +2,147,483,647 |

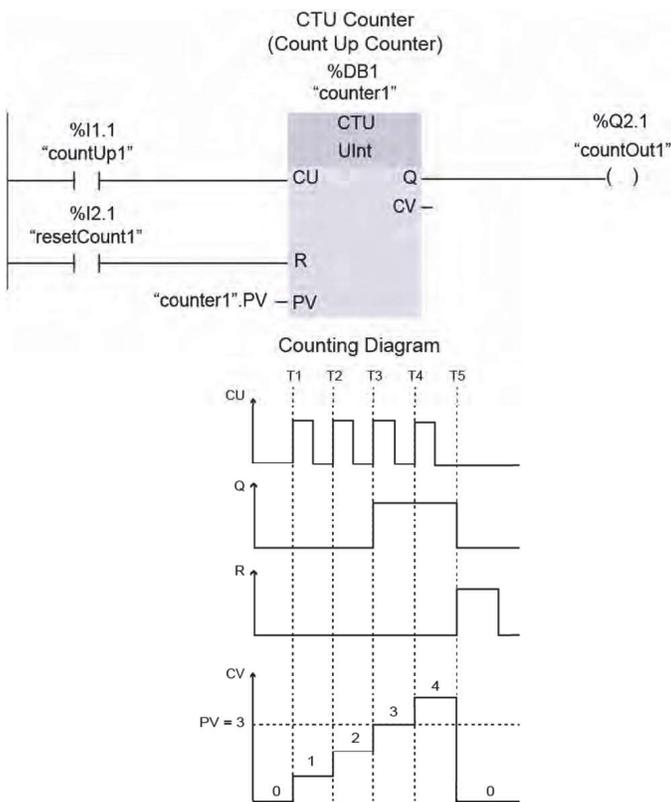
For LAD programs, S7-1200 PLCs support the IEC counter instructions shown in the accompanying graphic.

CU, CD, and CTUD counter instructions use software counters with a maximum counting rate that is limited by the execution rate of the organization block (OB) they are placed in. S7-1200 PLCs also provide high-speed counters for counting events that occur faster than can be handled by IEC counters.

Each IEC counter instruction uses a data block (DB) to maintain counter data. The number of counter instructions that you can use in a user program is limited only by the amount of memory in the CPU.

The amount of memory required for an instruction depends on the data type selected to store count values; therefore, it makes sense to choose the data type that matches the application requirements. The accompanying graphic also shows the data types available for IEC counters in S7-1200 PLCs.

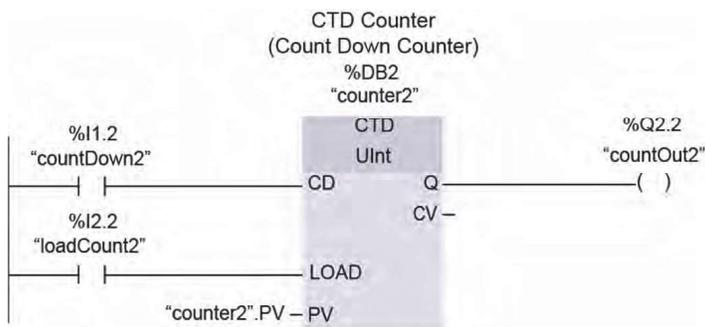
# CTU Counter Instruction



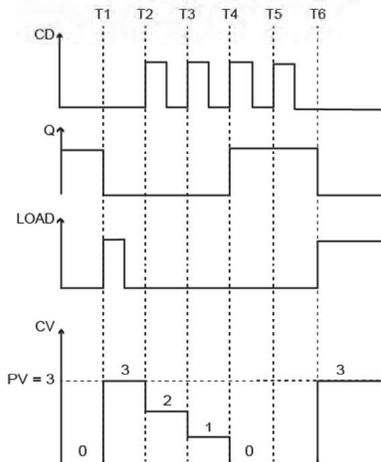
A CTU counter, also called a count up counter, is an instruction that counts up with each 0 to 1 transition of the count up parameter (CU). When the count value parameter (CV) reaches the preset value (PV), the output parameter (Q) transitions from 0 to 1. CV continues to increment with each 0 to 1 transition of CU until the high limit of the data type selected is reached. When the reset parameter (R) transitions from 0 to 1, CV and Q are reset to 0.

The accompanying graphic shows a CTU counter in a network and an associated counting diagram. In this example, PV is equal to 3. Each closure of the I1.1 contact causes a 0 to 1 transition of CU, which increments CV. At time T3, when CV equals PV, Q and output Q2.1 transition from 0 to 1. The count continues at time T4 when CV increments to 4. Because the data type selected for the counter in this example is unsigned integer (UInt), it could count up to 65,535. At time T5, when the I2.1 contact closes, R transitions from 0 to 1, and CV, Q, and output Q2.1 are reset to 0.

# CTD Counter Instruction



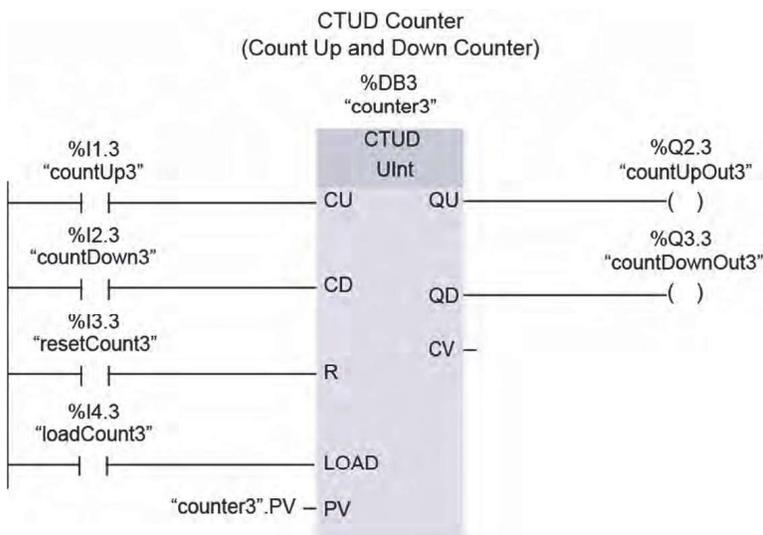
Counting Diagram



A CTD counter, also called a count down counter, is an instruction that counts down with each 0 to 1 transition of the count down parameter (CD). When the LOAD parameter transitions from 0 to 1, the count value parameter (CV) is set equal to the preset value (PV), and the output parameter (Q) is reset to 0. Each 0 to 1 transition of CD decrements CV by 1. When CV reaches 0, Q is set to 1. CV continues to decrement by 1 with each 0 to 1 transition of CD until the low limit of the data type is reached.

The accompanying graphic shows a CTD counter in a network and an associated counting diagram. In this example, PV is equal to 3. At time T1, the I2.2 contact closes causing a 0 to 1 transition of LOAD, which sets CV to 3 and resets Q and output Q2.2 to 0. Prior to time T2, the I2.2 contact opens, and LOAD is reset to 0. Thereafter, each closure of the I1.2 contact causes a 0 to 1 transition of CD, which decrements CV. At time T4, when CV equals 0, Q and output Q2.2 transition from 0 to 1. Because the data type selected for the counter in this example is unsigned integer (UInt), it stops counting down at 0. At time T6, when the I2.2 contact closes, LOAD transitions from 0 to 1, CV is set equal to 3, and Q and output Q2.2 are reset to 0.

## CTUD Counter Instruction



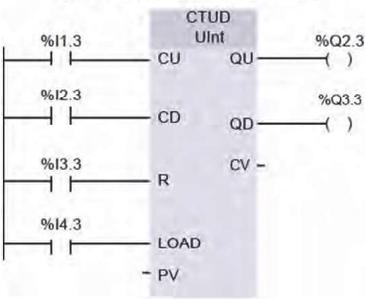
A CTUD counter, also called a count up and down counter, is an instruction that counts up with each 0 to 1 transition of the count up parameter (CU) and counts down with each 0 to 1 transition of the count down parameter (CD). If CU and CD both transition from 0 to 1 at the same time, the count does not change.

The current value of the count is available at the count value parameter (CV). The maximum and minimum values of CV and the present value parameter (PV) are determined by the data type selected for the counter.

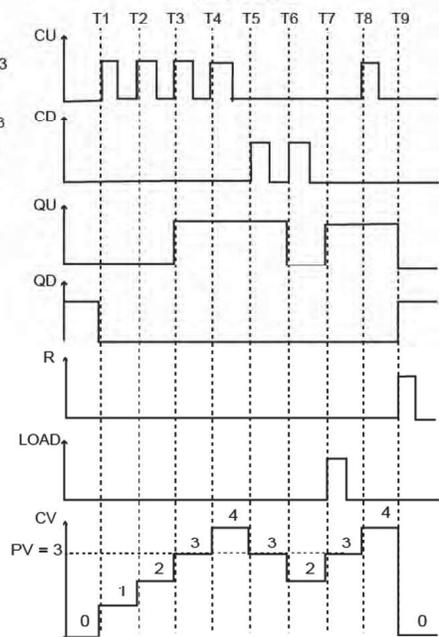
A 0 to 1 transition of the LOAD parameter sets CV equal to PV. A 0 to 1 transition of the reset parameter (R) resets CV to 0. QU is equal to 1 if CV is greater than or equal to PV. QD is equal to 1 if CV is less than or equal to 0.

# CTUD Counter Instruction Example

CTUD Counter  
(Count Up and Down Counter)



Counting Diagram



The accompanying graphic shows a CTUD counter in a network. For this example, PV is equal to 3. The counting diagram shown in the accompanying graphic is for the following sequence of events.

- At the start of the sequence, CV is 0, so QD and output Q3.3 are 1, and QU and output Q2.3 are 0.
- At time T1, the I1.3 contact closes, CU transitions from 0 to 1, and CV is incremented to 1. Because CV is greater than 0, QD and output Q3.3 are reset to 0.
- Prior to time T2, the I1.3 contact opens. At time T2, the I1.3 contact closes, CU transitions from 0 to 1, and CV increments to 2.
- Prior to time T3, the I1.3 contact opens. At time T3, the I1.3 contact closes, CU transitions from 0 to 1, and CV increments to 3. Because CV is equal to PV, QU and output Q2.3 are set to 1.
- Prior to time T4, the I1.3 contact opens. At time T4, contact I1.3 closes, CU transitions from 0 to 1, and CV increments to 4.
- At time T5, the I2.3 contact closes, CD transitions from 0 to 1, and CV decrements to 3.
- Prior to time T6, the I2.3 contact opens. At time T6, contact I2.3 closes, CD transitions from 0 to 1, and CV decrements to 2. Because CV is now less than PV, QU and output Q2.3 are reset to 0.
- At time T7, the I4.3 contact closes, LOAD is set to 1, CV is set equal to PV, and QU and output Q2.3 are set to 1.
- Prior to time T8, contacts I1.3 and I4.3 open. At time T8, the I1.3 contact closes, CU transitions from 0 to 1, and CV is incremented to 4.
- At time T9, the I3.3 contact closes, R, QD, and output Q3.3 are set to 1, and CV, QU, and output Q2.3 are reset to 0.