AMCI Frequently Asked Question

How Do I Generate "Velocity At Zero" and "Motion Direction" Bits In PLC Ladder Logic?

Many applications that use resolvers require outputs that turn off when the machine is not in motion or outputs that fire only in one direction. In order to make programming easier in these applications, AMCI has begun to add a "Velocity at Zero" and "Motion Direction" bits to our products as they are released or revised. This FAQ shows you how to replicate these bits with PLC ladder logic if your module does not yet offer these features.

Velocity at Zero Bit Description

This bit is latched when the change in position between scans becomes equal to or less than a *Low Threshold Value* and is unlatched when the change in position between scans exceeds a *High Threshold Value*. The latch and unlatch instructions are used to build hysteresis into the bit's logic. In order for the hysteresis to work correctly, the High Threshold Value must be at least one greater than the Low Threshold Value. Without this hysteresis, the Velocity at Zero bit would turn on and off intermittently when the speed of the resolver falls between a *Minimum Threshold Speed* and *Maximum Threshold Speed*. These Threshold Speeds are determined by the Scale Factor programmed into the module, the scan time of the processor, and the numbers you enter for your Low and High Threshold Values.

When decelerating, the *Minimum Threshold Speed* is the speed at which the Velocity at Zero Bit turns on. The bit remains on until the speed later increases above the *Maximum Threshold Speed*. When accelerating, the *Maximum Threshold Speed* is the speed at which the Velocity at Zero Bit turns off. The bit remains off until the speed later decreases below the *Minimum Threshold Speed*.

The formula for calculating the Threshold Speeds is given below. Note that the scan time must be expressed in seconds.

$$\text{High/Low Threshold Speed}_{\text{RPM}} = \frac{60 \times (\text{High/Low Threshold Value} + 1)}{\text{Scale Factor} \times \text{Scan Time}_{\text{seconds}}}$$

For example, assume a Low Threshold Value of zero, a Scale Factor of 1000 and a Scan Time of 10 milliseconds. The Minimum Threshold Speed will be: $(60 \times (0+1)) / (1000 \times 0.01) = 6.00$ RPM. If your Scale Factor is instead 360, the Minimum Threshold Speed would be 16.67 RPM. If your Scale Factor is 8,192, the Minimum Threshold Speed would be 0.73 RPM.

If the default High Threshold Value of "1" is used with the default Low Threshold Value of "0", the High Threshold Speed will be double the Low Threshold Speed. If the minimum speed you calculate is too great for your application, then you have to accumulate a change in position over multiple scans or place the ladder logic in a timed interrupt. If you need help in developing this type of ladder logic, contact AMCI for assistance.

Motion Direction Bit Description

The Motion Direction Bit is off when the position value is increasing and on when the position value is decreasing. It remains in its last state when there is no change in position during the scan. In order for the Motion Direction bit to work correctly, the resolver cannot complete more that 1/2 of a turn in one PLC scan. Therefore, your maximum speed, in RPM, is equal to:

$$\text{Maximum Speed}_{\text{RPM}} = \frac{30}{\text{Scan Time}_{\text{seconds}}}$$

For example, assume you maximum scan time is 20 milliseconds. Your resolver cannot exceed: $(30/0.02) = 1500$ RPM.

> The ladder logic is written for Allen-Bradley PLC's. However, the code can be converted to any processor system that AMCI manufactures product for.

Memory Needed

The ladder logic requires eight words of memory and an additional two bits of memory as the Motion Direction and Velocity at Zero bits. The word memory should be *signed integer words* as it is the most efficient and supported by all processors. Five of the eight words are used to store constants for limit tests and are not strictly necessary, because they can be programmed as literals into the instructions. However, storing them in memory makes it easier to change your setup at a later time.

- **N7:0: Current Position.** This is the position data read from the AMCI module this scan.
- **N7:1: Previous Position.** This is the position data read from the module during the previous scan.
- **N7:2: Δ Position.** The change in position between two consecutive scans.
  Δ Position = (Current Position – Previous Position).
- **N7:3: Scale Factor.** This is the Scale Factor value programmed into the module. You can save a memory location by programming this value as a literal into the instructions that use it, but using a memory location to hold the value makes it easier to change your code later.
- **N7:4: Negative Zero Cross Compare Value.** This value is used to determine if the resolver has crossed through its zero point, and is equal to: -1 × (Scale Factor / 2). You can save a memory location by placing this value as a literal in the instruction that uses it, but using a memory location to hold the value makes it easier to modify your code later.
- **N7:5: Positive Zero Cross Compare Value.** This value is used to determine if the resolver has crossed through its zero point, and is equal to: (Scale Factor / 2). You can save a memory location by placing this value as a literal in the instruction that uses it, but using a memory location to hold the value makes it easier to modify your code later.
- **N7:6: Low Threshold Value.** This value is used to determine if the Velocity at Zero bit should be latched and has a default value of zero. You can save a memory location by placing this value as a literal in the instruction that uses it, but using a memory location to hold the value makes it easier to modify your code later.
- **N7:7: High Threshold Value.** This value is used to determine if the Velocity at Zero bit should be unlatched and has a default value of one. It must be at least one greater than the Low threshold Value. You can save a memory location by placing this value as a literal in the instruction that uses it, but using a memory location to hold the value makes it easier to modify your code later.
- **B3:0/00: Velocity at Zero Bit.** This bit is set when the change in position between scans falls below the Low Threshold Value and is reset when the change in position is greater than the High Threshold Value. The bit stays in its last state when the change in position falls between teh Low and High Threshold Values.
- **B3:0/01: Motion Direction Bit.** This bit is reset when the position count is increasing and set when the position count is decreasing. This bit remains in its last state when there is no change in position values between scans.

Ladder Logic

Motion_Bits.rss

LAD 2 - --- Total Rungs in File = 10

Buffer the Position Value to make sure it does not change during the scan. In this example. N9:0 contains the data from the module and N7:0 is used to buffer the data.

0000

Buffered Position
MOV
Move
Source N9:0
27<
Dest N7:0
27<

Calculate the difference between the current scan position data, N7:0, and the previous scan position data, N7:1. The difference between these two values is stored in N7:2.

0001

Current Position -
Previous Position
SUB
Subtract
Source A N7:0
27<
Source B N7:1
355<
Dest N7:2
-328<

If the difference in position between the two scans is less than negative 1/2 of the Scale Factor, then the resolver has traveled through its zero point and the position data is increasing. If this occurs, add the Scale Factor to the difference stored in N7:2 to compensate for the zero crossing. In this example, the Scale Factor equals 360, therefore, the negative of half the Scale Factor is -180.

0002

Current Position -
Previous Position
LES
Less Than (A<B)
Source A N7:2
-328<
Source B N7:4
-180<

Current Position -
Previous Position
ADD
Add
Source A N7:2
-328<
Source B N7:3
360<
Dest N7:2
32<

If the difference in position between the two scans is greater than 1/2 of the Scale Factor, then the resolver has traveled through its zero point and the position data is decreasing. If this occurs, subtract the Scale Factor from the difference stored in N7:2 to compensate for the zero crossing. In this example, the Scale Factor equals 360, therefore, half of the Scale Factor is 180.

0003

Current Position -
Previous Position
GRT
Less Than (A>B)
Source A N7:2
32<
Source B N7:4
180<

Current Position -
Previous Position
SUB
Subtract
Source A N7:2
32<
Source B N7:3
360<
Dest N7:2
32<

20 Gear Drive, Plymouth Ind. Park, Terryville, CT 06786
Tel: (860) 585-1254 Fax: (860) 584-1973 http:\\www.amci.com

Ladder Logic (continued)

Motion_Bits.rss

LAD 2 - --- Total Rungs in File = 10

If the difference in position between scans is less than or equal to the Low Threshold Value, latch the Velocity at Zero Bit.

0004
Current Position - Previous Position
LEQ
Less Than Or Equal
Source A N7:2
32<
Source B N7:6
0<
Velocity At Zero Bit
0 = motion
1 = no motion
B3:0
L
0

If the difference in position between scans is greater than the High Threshold Value, unlatch the Velocity at Zero Bit.

0005
Current Position - Previous Position
GTR
Greater Than (A>B)
Source A N7:2
32<
Source B N7:7
1<
Velocity At Zero Bit
0 = motion
1 = no motion
B3:0
U
0

If the difference in position between scans is negative, latch the Motion Direction Bit to indicate that the resolver's position data is decreasing. The latch instruction is used so that the bit remains in its last state when there is no motion.

0006
Current Position - Previous Position
LES
Less Than (A<B)
Source A N7:2
32<
Source B 0
0<
Motion Direction Bit
0 = increasing
1 = decreasing
B3:0
L
1

If the difference in position between scans is positive and non-zero, unlatch the Motion Direction Bit to indicate that the resolver's position data is increasing. The unlatch instruction is used so that the bit remains in its last state when there is no motion.

0007
Current Position - Previous Position
GTR
Greater Than (A>B)
Source A N7:2
32<
Source B 0
0<
Motion Direction Bit
0 = increasing
1 = decreasing
B3:0
U
1

Store the Current Position data in register N7:1 for use during the next program scan as the Previous Position data.

0008
Buffered Position
MOV
Move
Source N7:0
27<
Dest N7:1
27<

0009
END

20 Gear Drive, Plymouth Ind. Park, Terryville, CT 06786
Tel: (860) 585-1254 Fax: (860) 584-1973 http:\\www.amci.com