7561 Manual Addendum

Determining the Format Parameters of an unknown sensor.

SSI data is typically twenty-five bits long. However, not all twenty-five of these bits are necessarily part of the data value. Some of the bits may be status bits or error bits. If your sensor does not have a data sheet showing which bits of the 25 bit data stream contain the SSI data value, than you can look at the raw SSI data in the input data words to determine which bits represent the data value. The raw SSI data is stored in input words 5 and 6, with the LSB in bit zero of word 6. As shown in the following figure, the bottom sixteen bits of a 25 bit transfer is stored in word 6, while bits 1 through 9 are stored in the bottom half of word 5.

0	0	0	0	0	0	0	Bit 1	Actual SSI Data	Bit 9	**Word 5**
Bit 10	Actual SSI Data								Bit 25	**Word 6**

The first parameters that you need to determine are the Number *of SSI Data Bits* and *MSB* parameters. These values should be easy to determine by running the transducer through its full range of motion and observing which bits change in the SSI data. However, be aware that the transducer may have status bits that change as the transducer runs through its full range. Once you determine the MSB and LSB of the data value, the difference between the two, plus one, is the number of data bits. Also, it is a good idea to observe the bit pattern while slowly moving the transducer to see if the data is in binary or gray code format.

Once the *Number of Data Bits* and *MSB* parameters have been determined, the last two parameters that you need to set are *Data Logic* and *Data Type*. Start by setting the Data Logic parameter to it default *Positive* value. Set the Data Type parameter to Binary unless you were able to determine that the data is in gray code format. Hopefully, the Data Value input words are stable at this point. In this case, stable means that it changes in a predictable manner. The Data Value still may not be the one you're looking for, but you should be able to solve these problems by applying the correct scalars, count direction, and presets. If the Data Value is not stable, follow these suggestions for correcting it.

1. Switch the value of the Data Type parameter. If this does not solve the problem, again look closely at the bit pattern in input words 5 and 6 to see if there are any status bits included in the Data Value. Also see if you can determine the data type from the bit pattern as you change it slowly.
2. If you're sure that the Data Type is binary but you have less then expected resolution, then you may be missing the LSB's of the Data Value. Increase the Number of Data Bits parameter.
3. If you're sure that the Data Type is gray code and the Data Value is still incorrect, then the problem usually lies in the *Number of Data Bits* and *MSB* parameters. You can pull a sub-set of binary coded data out and it will still be correct, but the conversion from gray code to binary format demands that you get the LSB of the data and all of the bits that change in SSI data. In this case, try changing the Number of Data Bits and MSB parameters.

Scalar Multiplier & Divisor

The resolution of the linear displacement transducers manufactured by Balluff and MTS have increased since the 7561 module's manual was released. The following table shows the appropriate Scalar Multiplier and Divisor values for all of the currently available LDT resolutions.

LDT Resolution	Desired Resolution						
	0.00005"	0.0001"	0.0002"	0.0005"	0.001"	0.002"	0.005"
1µm	100/127	50/127	25/127	10/127	5/127	5/254	1/127
2µm		100/127	50/127	20/127	10/127	5/127	2/127
5µm			125/127	50/127	25/127	25/254	5/127
10µm				100/127	50/127	25/127	10/127
20µm					100/127	50/127	20/127
40µm						100/127	40/127

Use the following procedure to calculate your Scalar and Divisor values if either your LDT Resolution or Desired Resolution does not appear in the above table

Conversion Factor: $\dfrac{\text{Desired Resolution (counts/inch)}}{\text{LDT Resolution (counts/inch)}}$

Step 1: Convert your LDT resolution from µm to inches. For example, you are using a sensor with 1µm resolution in your application.

$$1\ \mu m \ * \ \frac{1\ mm}{1000\ \mu m} \ * \ \frac{1\ inch}{25.4\ mm} = 0.00003937\ \text{inches/count} = 25400\ \text{counts/inch}$$

Step 2: Determine the number of counts per inch for the desired resolution. For example, 0.0001".

0.0001 inch/count = 10000 counts/inch

Step 3: Determine the Scalar Multiplier and Divisor values.

$$\frac{\text{Desired Resolution (counts/inch)}}{\text{LDT Resolution (counts/inch)}} = \frac{10000\ \text{counts/inch}}{25400\ \text{counts/inch}} = \frac{100}{254} = \frac{50}{127}$$

Therefore, to use a sensor with 1µm resolution and get 0.0001 inches per count resolution, use a Scalar Multiplier of 50 and a Scalar Divisor of 127.

Reversing the Count Direction Ladder Logic Program

There are some errors on the ladder logic program shown on page 8 of the user manual. This program shows how to reverse the direction of increasing counts of a rotary encoder. The corrected program is shown below.

I:1.1 & I:1.2: Holds the Data Value from the 7561. Assumes that the 7561 is in slot 1.
N7:20: Holds the maximum number of encoder counts. This is 3,600 in this example.
N7:21: Holds the combined Data Value.
N7:22: Holds the reversed Encoder Count Value.

Because the Data Value can be greater than 32,767, it is transmitted to the processor in two words. This rung combines the two words and stores the value in N7:21. Also note that the maximum encoder counts supported with this logic is 32,767. If your encoder value will exceed 32767, floating point registers will have to be used in place of registers N7:20 through N7:22.

Reverse the count direction by subtracting the combined Data Value, N7:21, from the number of encoder counts that you stored in N7:20, and store the result in N7:22. This example assumes that a 3,600 count encoder is being used.

The zero count of the encoder is the same in both directions. Therefore, if the combined Data Value in N7:21 is zero, move a value of zero into the reversed count, N7:22. Register N7:22 is now ready to be used as the position value by the rest of the program.

Notes on Rate of Change Data

The 7561 module calculates the Rate of Change, which is velocity in rotary application, as the change in position over the programmed update time. Whenever a rotary encoder rolls over from its maximum to minimum value or vice versa, the 7561 will see a very large change in position in one update time calculation. If you are using the velocity data from the 7561, you should first calculate the maximum speed of your machine and write some ladder logic to ignore any velocity data from the 7561 that exceeds this value.

The Rate of Change Data is reported in units of Counts / Second. The 7561 module has a programmable Rate Update Time of 1 to 1000ms. This combination, especially if a smaller rate update time is used, will result in a rate value where some of the reported digits are zero. This is illustrated by the following two examples.

Example 1

Rate Update Time = 1ms
Rate data from sensor = 12345 counts / second

In this case, the 7561 would report a rate value of 12000.

Example 2

Rate Update Time = 10ms
Rate data from sensor = 12345 counts / second

In this case, the 7561 would report a rate value of 12300.

File: 7561_manual_addendum.doc
Date: 11/18/05