PowerFlex 700S DriveLogix

Allen-Bradley

Tech Tips

Using The 20-COMM-D

Description

This document serves as a supplement to the 20-COMM-D User Manual (20COMM-UM002x) addressing items specific to the PowerFlex 700S. Please refer to the 20-COMM-D User Manual for details on 20-COMM-D set-up, configuration, I/O messaging, and explicit messaging.

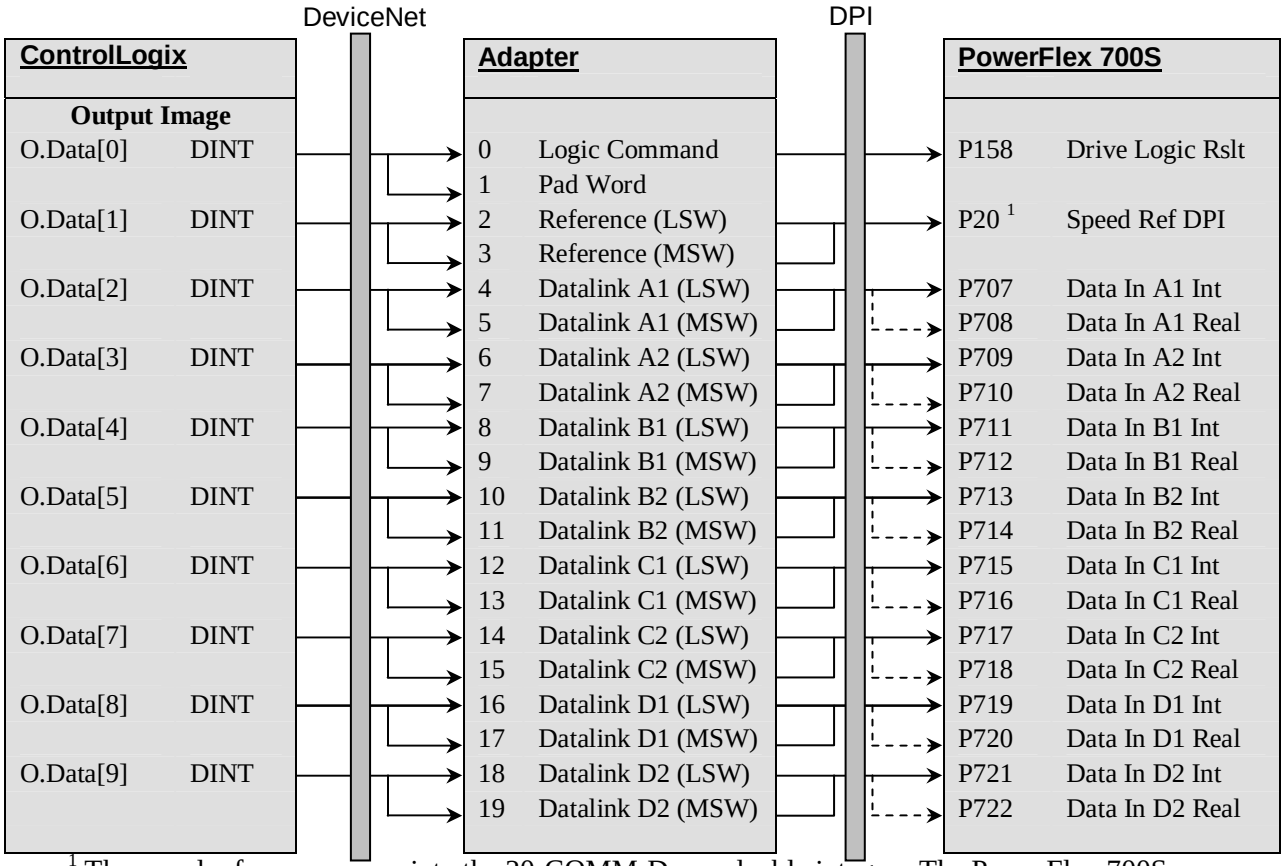
Technical Information

To use the 20-COMM-D with the PF700S, the 20-COMM-D must be v1.005 firmware or later.

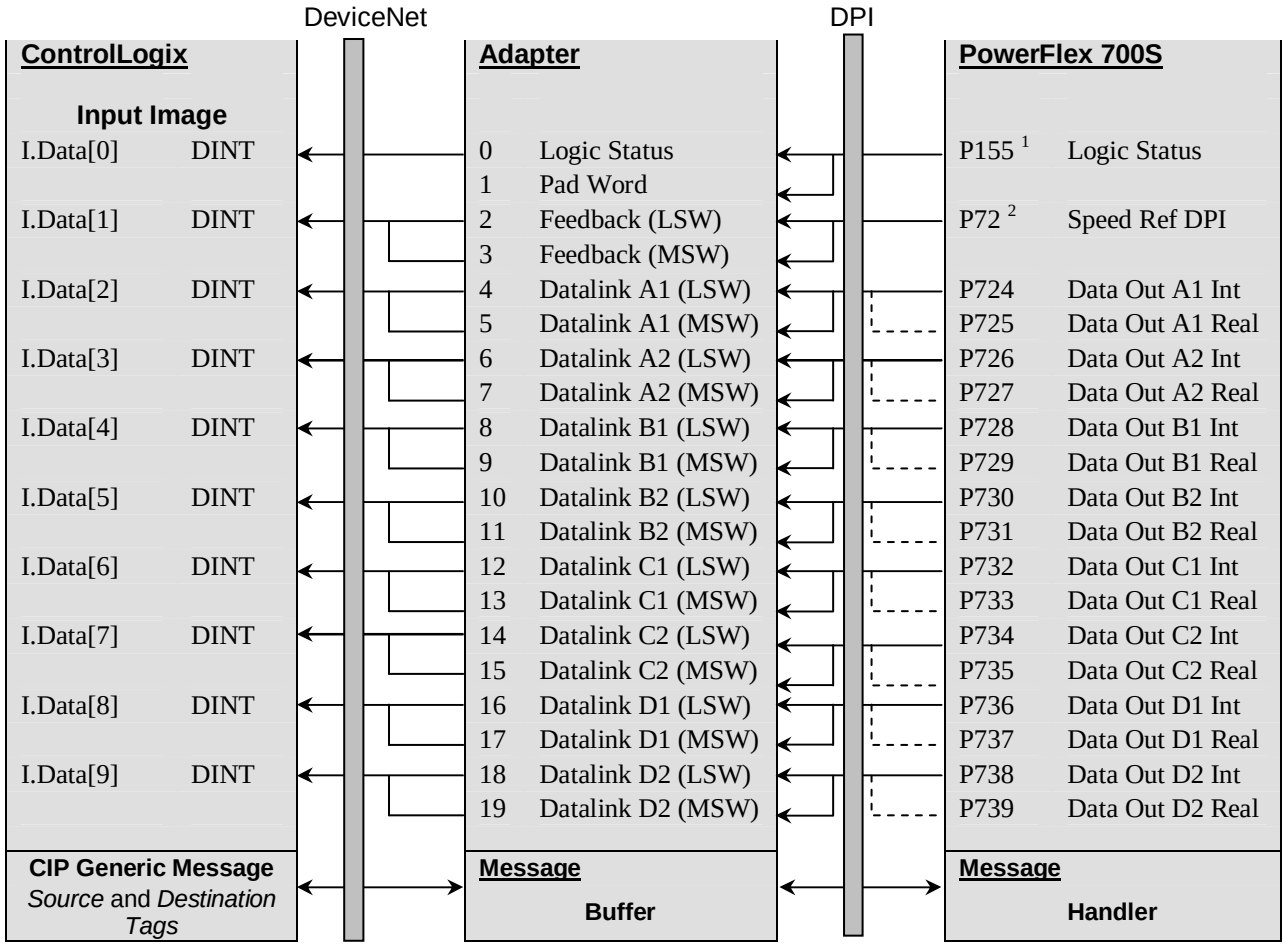
The Logic Command and Logic Status are 16 bits plus a 16 bit pad word for a total of 32 bit data. The bit definitions of the Logic Command word follow the same pattern as P158 [Drive Logic Rslt]. The bit definitions of the Logic Status word follow the same pattern as bits 0-15 of P155 [Logic Status]. Reference, Feedback, and Datalinks are 32 bit data. This means with just the Logic Command/Status and Speed Ref/Fdbk I/O enabled in the 20-COMM-D, we would map 8 bytes of I/O in the DeviceNet Scanner. With the Logic Command/Status, Speed Ref/Fdbk and all of the Datalinks enabled we would have a total of 40 bytes of I/O mapped in the DeviceNet scanner.

ControlLogix System

Here is the I/O Image table for a ControlLogix system.



Notes: [1] The speed reference comes into the 20-COMM-D as a double integer. The PowerFlex 700S firmware automatically converts that speed reference into floating point, so that P20 [Speed Ref DPI] is a floating point value.



Notes: [1] Bits 0-15 only of P155 [Logic Status] appear in the Input Image table of the ControlLogix controller.
[2]The speed feedback sent from the PowerFlex 700S to the 20-COMM-D is not affected by P73 [Spd Fdbk Scale]. Furthermore, the PowerFlex 700S automatically coverts P72 [Scaled Spd Fdbk], which is a floating point parameter, to a double integer before the value is transferred to the 20-COMM-D.

Reference/Feedback Programming
To setup the PowerFlex 700S to follow a speed reference from the 20-COMM-D, P691 [DPI Ref Select] must be set to "Port 5". P16 [Speed Ref Sel] must be set to "Speed Ref DPI".

Reference and Feedback values are floating point values in the PowerFlex 700S. Use the following logic to transmit and receive reference and feedback data as integer data.

Transmitted Reference = [Floating point Reference (RPM) * 32768] / [Base motor RPM]
Floating point Feedback (RPM) = [Feedback received * Base motor RPM] /32768

Datalink Programming
To read datalinks, the bits in P723 [Dlink OutDataTyp] must be set appropriately for each Datalink to select whether the data is floating point or DINT.

In the ControlLogix system, Datalinks are transmitted over DeviceNet as 32 bit integers (DINT). In order to send or receive floating point a COP (copy) instruction must be utilized. The copy instruction in ControlLogix performs a bitwise copy. Set the length of the copy instruction to a value appropriate for the destination data type. For example, when copying a DINT data type to a REAL data type, the length would be one since both data types contain 32 bits of data.

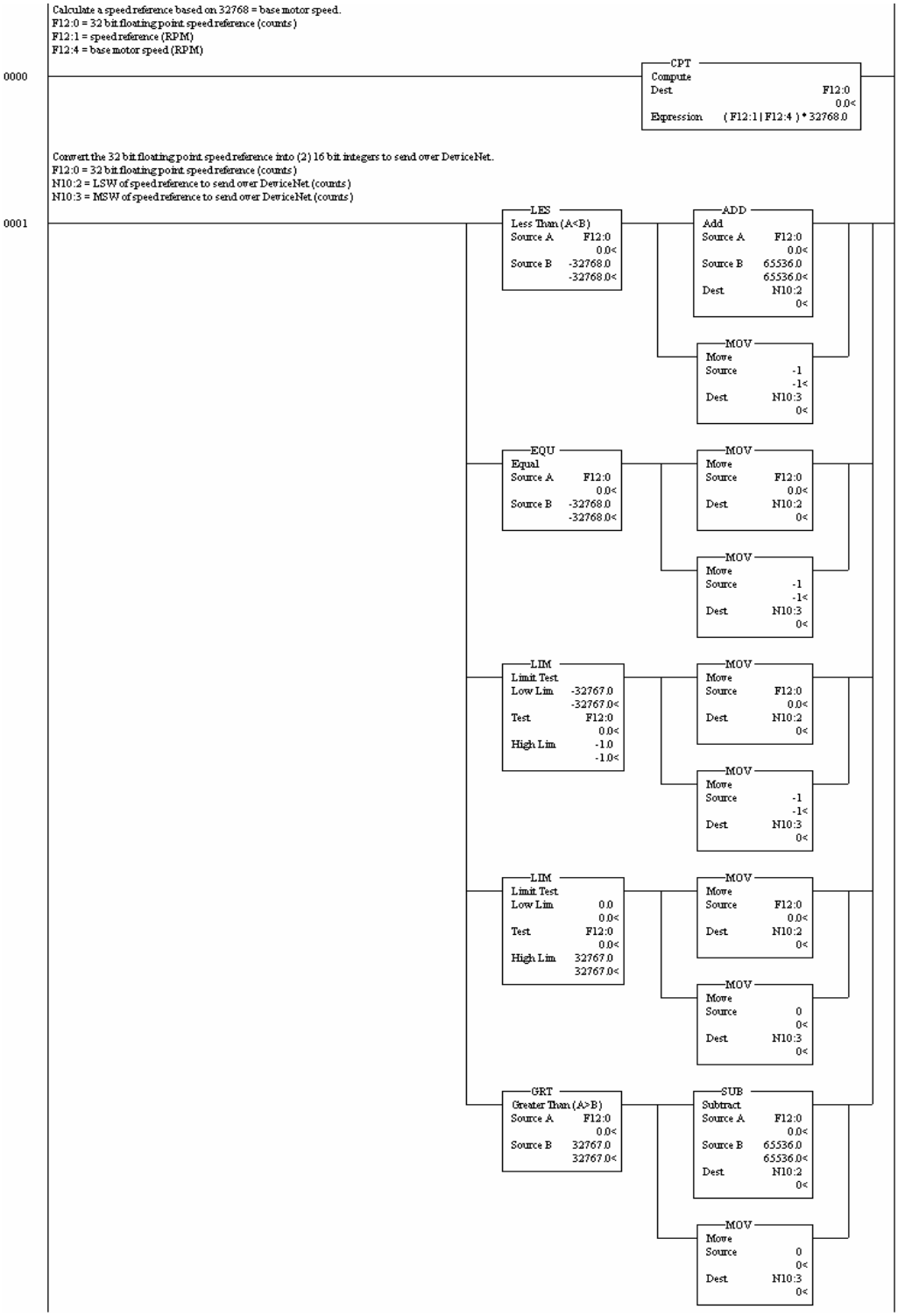
Explicit Message Programming

When using explicit messaging in the ControlLogix system, the message type CIP Generic is used. The data is transferred over DeviceNet in the same data type as the parameter in the PowerFlex 700S. Make sure that the data type for the *Source* and *Destination* tags in your ControlLogix message instruction matches the data type in the PowerFlex 700s. Also, the *Num. of Elements* in the ControlLogix message instruction must match the size of the Source data. For example, to send an explicit message to write to P12 [Speed Ref 2], which is a floating point:

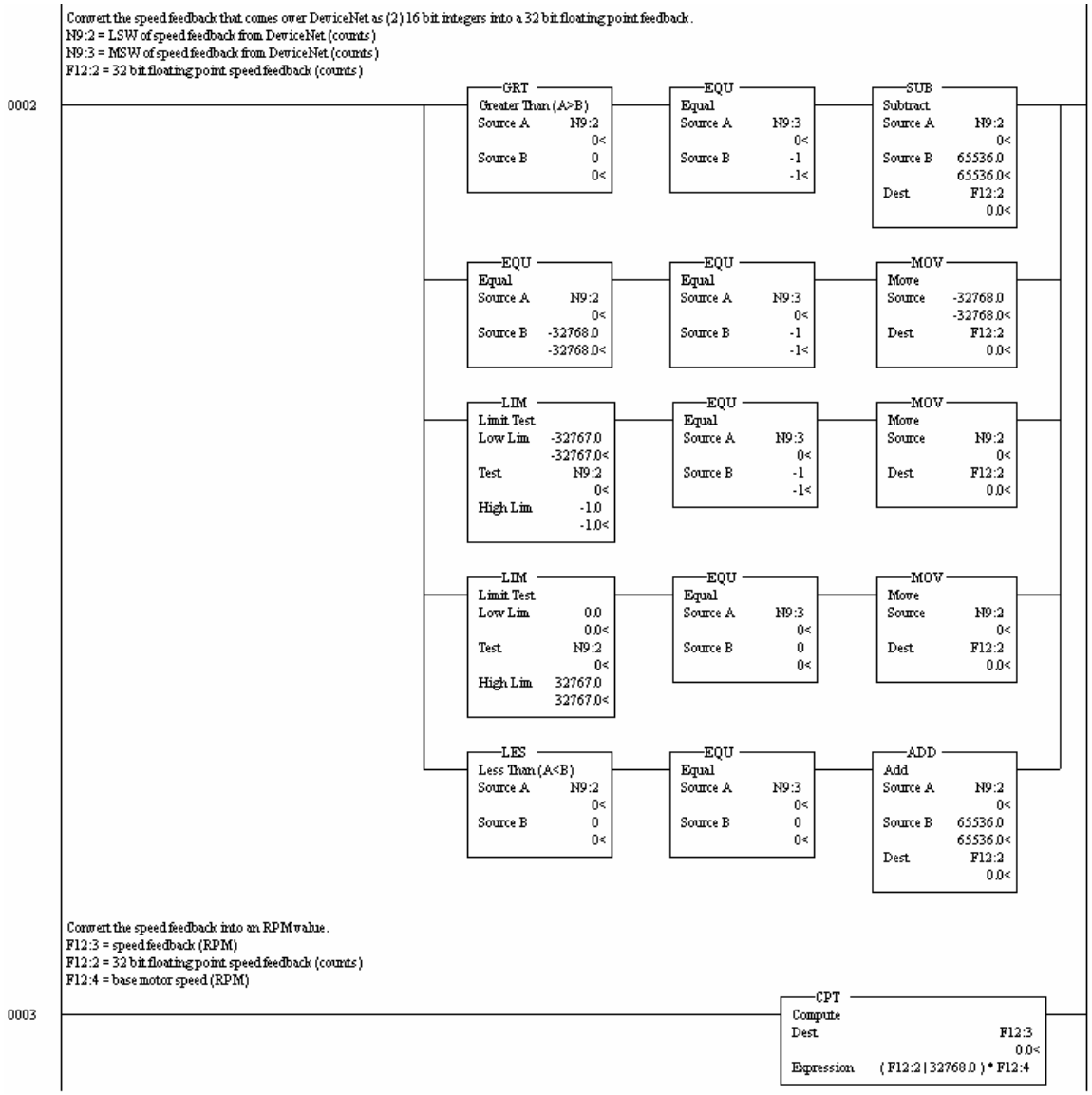
1. The *Source* and *Destination* tags would be of type REAL.
2. The *Num. of Elements* would be 4 bytes since a REAL data type takes up 4 bytes of data.

SLC/PLC-5 System Reference/Feedback Programming

The reference is scaled so that base motor speed = 32768. The SLC/PLC-5 does not use DINT, and only handle 16 bit integers, so the reference has to be handled differently to account for references above 32767 or below -32768. The following example shows how to transmit references less than twice base motor speed.



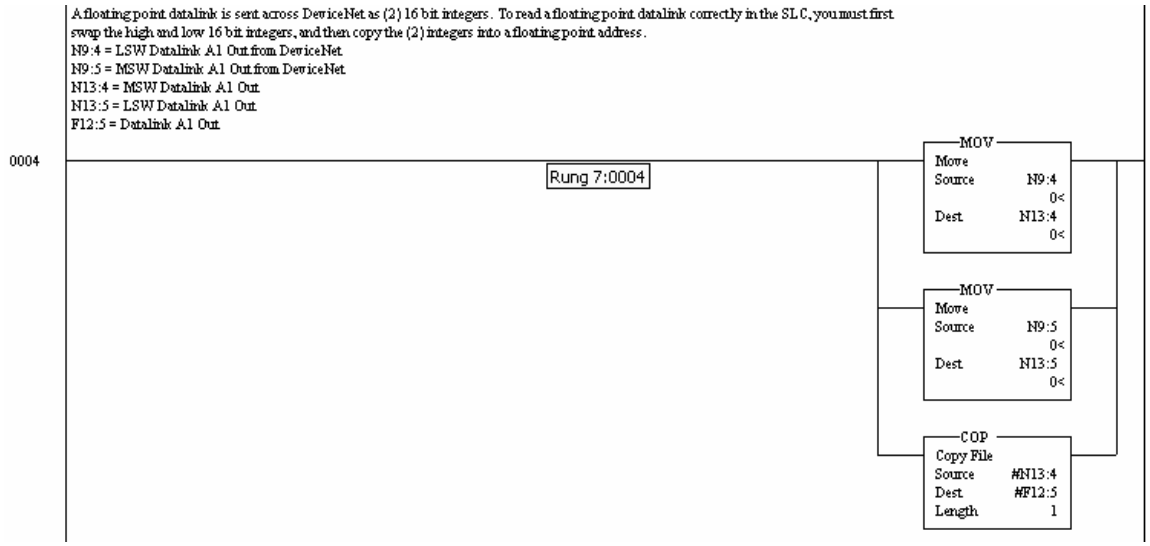
The feedback is also scaled so that base motor speed = 32768. The SLC/PLC-5 does not use DINT, and only handle 16 bit integers, so the feedback has to be handled differently to account for references above 32767 or below -32768. The following example shows how to read feedback values less than twice base motor speed.



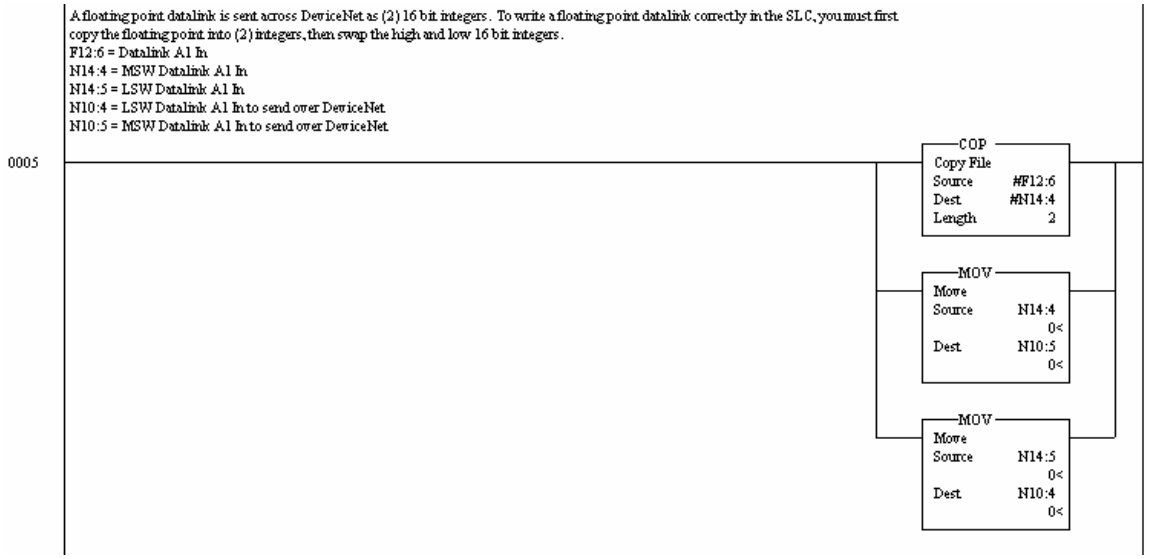
Datalink Programming

Datalinks are transmitted and received through block transfer I/O. The SLC/PLC-5 is limited to 16 bit integers and floating point. Because the SLC/PLC-5 does not support 32-bit integers, 32-bit integer Datalinks remain split into (2) 16 bit integers. In order to send or receive floating point Datalinks we have to swap the LSW and MSW and utilize the COP (copy) instruction. The following examples are for transmitting and receiving the floating-point Datalinks.

Reading Floating-Point Datalinks in an SLC/PLC-5.



Writing Floating-Point Block Datalinks in an SLC/PLC-5.

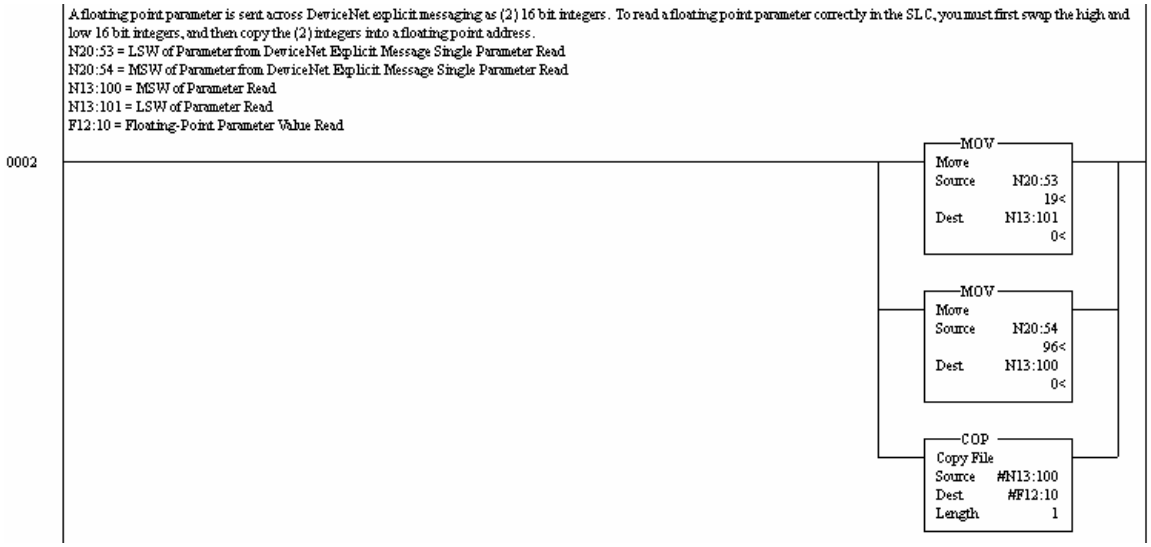


Explicit Messaging

Explicit block transfer messaging is used to configure the drive and monitor data from the drive. This type of block transfer is different than the block transfer used to transmit and receive datalinks. Chapter 6 of the 20-COMM-D User Manual shows the format of the explicit message request and response data in an SLC and PLC-5.

Because the SLC/PLC-5 does not support 32-bit integers, 32-bit integer data from the explicit message request and response data remains split into (2) 16 bit integers. In order to send or receive floating point data we have to swap the LSW and MSW and utilize the COP (copy) instruction. The following examples are for transmitting and receiving floating-point data for explicit messages.

Reading Floating-Point Explicit Message Data in an SLC/PLC-5.



Writing Floating-Point Explicit Message Data in an SLC/PLC-5.