

# SLC 500 Ladder Instructions For SoftPLC

## I. INTRODUCTION

The SLC 500 PLC has many ladder instructions that are not part of SoftPLC's instruction set. There are also some basic ladder instructions, like multiply (MUL) and divide (DIV) that operate differently. As a result, the SLC loadable Module (SLC.TLM) was created to implement some of these unique instructions as an extension to the already rich SoftPLC instruction set. There is a name for this type of extension ladder instruction; it is called a "TOPDOC Loadable Instruction" or simply a "TLI".

## II. SOFTWARE INSTALLATION

The "SLC.TLM" must be installed on *SoftPLC's* flash disk in the "\SoftPLC\tlm" directory. This can be done using a FTP Client such as "Filezilla". Filezilla may be installed from the *SoftPLC* Product CD.

In order for the driver to be selected in the TOPDOC NexGen's Module Editor, the SLC.DEF file must be installed on the Windows PC in the "\SoftPLC\PLC" directory.

For off-line programming with TOPDOC NexGen, the SLC.TLM must be installed on the Windows PC "\SoftPLC\tlm" directory. A local PLC definition must also be created that has the SLC.TLM "Checked for Use" in the Module Editor.

**MODULE=C:\SoftPLC\tlm\SLC.TLM**

## IV. SLC INSTRUCTIONS IMPLEMENTED IN "SLC.TLM"

The following eleven (11) SLC instructions have been implemented as TLI's for use with SoftPLC. All of the TLI's in this module are "Output" Instructions.

1. **ABS**                    **"SLC Absolute Value"** - Determines Absolute value of the Source parameter. Either parameter can be a Floating Point or Integer value.

Param0 - Source            <Value or address>  
Param1 - Dest              <Must be address!>

2. **DCD**                    **"SLC Decode 4 to 1 of 16"** - Decodes 4 bit value and turns on corresponding bit in Destination word. Sixteen (16) bit Integer data types only!

Param0 - Source            <Integer value or address>

Param1 - Dest <Must be address!>

3. **DDV** **“SLC Double Divide”** - The 32 bit content of the math register (S:6-7) is divided by the 16 bit “Source” value and the rounded quotient is placed in the destination (Dest). If the remainder is 0.5 or greater, the destination is rounded up. This instruction typically follows a MUL instruction that creates a 32 bit result which would reside in the math register. The math status register (S:0) is also updated with overflow, zero and sign status. If an overflow occurs the minor error bit is also set (S:10/14)

Param0 - Source <integer value or address>  
Param2 - Dest <Must be address!>

4. **DIV** **“SLC Divide”** - Divide source A (Src A) by source B (Src B). The rounded quotient is placed in destination (Dest). If the remainder is 0.5 or greater, the result in “Dest” is rounded up. The unrounded quotient is placed in the most significant word of the math register (S:7). The remainder is placed in the least significant word of the math register (S:6). The math status register (S:0) is also updated with overflow, zero and sign status. If an overflow occurs the minor error bit is also set (S:10/14)

Param0 - Src A <integer or float value or address>  
Param1 - Src B <integer or float value or address>  
Param2 - Dest <integer or float address>

5. **ENC** **“SLC Encode 1 of 16 to 4”** - Decodes single set bit in Source to a 4 bit integer value (0-15). If more than one bit is set in Source, Set Overflow bit (S:0/1). If Source is zero, set Zero Flag in S:0/2.

Param0 - Source <Integer value or address>  
Param1 - Dest <Must be address!>

6. **MUL** **“Slc Multiply”** - Multiple source A (Src A) by source B (Src B) and place the result in the destination (Dest). If an overflow occurs, the 32 bit signed result is placed in the math register S:6 (LSW) and S:7 (MSW). The math status register (S:0) is also updated with carry, overflow, zero and sign status. If an overflow occurs the minor error bit is also set (S:10/14)

Param0 - Src A <integer or float value or address>  
Param1 - Src B <integer or float value or address>  
Param2 - Dest <integer or float address>

7. **SCP** **“SLC Scale with Parameters”** - Scale Input value using givens and place in Output parameter. All values may be floating point or integers.

Param0 - Input  
Param1 - InputMin  
Param2 - InputMax  
Param3 - ScaleMin  
Param4 - ScaleMax  
Param5 - Output <Must be address!>

8. **SCL** ***“SLC Scale Data”*** - Scale Source using givens and place in Destination parameter. All values may be floating point or Integers.

Param0 - Source  
Param1 - Rate\*10K  
Param2 - Offset  
Param3 - Dest <Must be address!>

9. **SQCW** ***“SLC Sequencer Compare Word”*** - Compare each word in file using mask to Source word. If equal, set found (FD) bit, when all words compared in File, set done (DN) bit.

Param0 - File <File, length=256>  
Param1 - Mask <Integer value or address>  
Param2 - Source <Integer value or address>  
Param3 - Control <Control Element>

10. **SQCF** ***“SLC Sequencer Compare File”*** - Compare each word in file using mask to Source file. If equal, set found (FD) bit, when all words compared, set done (DN) bit.

Param0 - File <File, length=256>  
Param1 - Mask <Integer value or address>  
Param2 - Source <File, length=256>  
Param3 - Control <Control Element>

11. **SWP** ***“SLC Swap Low/High Bytes”*** - Swap low and high bytes in each 16 bit word of Source file.

Param0 - Source <File, length=128>  
Param1 - Length <Integer value or address>