



Open Architecture Control Software

USING SoftPLC WITH THE INDRAMAT CLC CARD

Because of the variety of uses of the information described in this application note, the users of, and those responsible for applying this information must satisfy themselves as to the acceptability of each application and use of the information. In no event will Tele-Denken Resources, Inc. be responsible or liable for its use, nor for any infringements of patents or other rights of third parties which may result from its use.

SoftPLC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE.

SoftPLC Corporation reserves the right to change product specifications at any time without notice.

No part of this document may be reproduced by any means, nor translated, nor transmitted to any magnetic medium without the written consent of SoftPLC Corporation.

SoftPLC and TOPDOC are registered trademarks of SoftPLC Corporation.

First Printing: April, 1996

(c) Copyright 1996 -2001 SoftPLC Corporation
ALL RIGHTS RESERVED

SoftPLC Corporation
25603 Red Brangus Drive
Spicewood, TX 78669
Telephone 512/264-8390 or 1/800-SoftPLC
FAX512/264-8399
Email: info@softplc.com
www.softplc.com

1. Introduction

SoftPLC employs a unique technology which lets a C programmer add new ladder logic instructions to the instruction set. These loadable instructions are called TLI's. TOPDOC, the ladder logic programming package which supports SoftPLC, automatically learns about the new TLI's as it logs into the SoftPLC. Additionally, simple entries may be placed into an ascii text file to inform TOPDOC about the TLI's so they may be used offline as well. Then the TLI's can even be used with complete access to the offline emulation and debugging capabilities of SoftWIRES.

TLI's may be developed by any competent C programmer who has access to the SoftPLC C Programmer's Toolkit, a product readily available from SoftPLC Corp. There are a number of Systems Integrators who are SoftPLC Partners who possess the requisite expertise. End users may also have this capability.

This document describes a number of TLI's, all which reside in a Loadable Module (TLM). The TLM is called **ICLC** and resides in a file with the name ICLC.TLM. This TLM was developed and is maintained by SoftPLC Corporation.

This TLM supports up to 16 CLC cards concurrently.

2. Installing the TLM

- 1) Copy the file ICLC.TLM into your \SPLCZ\IODVR directory on your SoftPLC machine.
- 2) With a text editor edit your \SPLCZ\MODULE.LST file and add the following line:

```
MODULE=\SPLCZ\IODVR\ICLC.TLM CARD0=DC00
```

If your card is located at a memory segment address other than DC00, then use the correct value, per the Indramat documentation and the positions of jumpers S8, S9, S10, and S11. See page 2-8 of the Rev. D Indramat CLC card user's manual. If you have more than one card, list them following the CARD0 entry as follows:

```
MODULE=\SPLCZ\IODVR\ICLC.TLM CARD0=DC00 CARD1=E400 CARD15=E000 ...
```

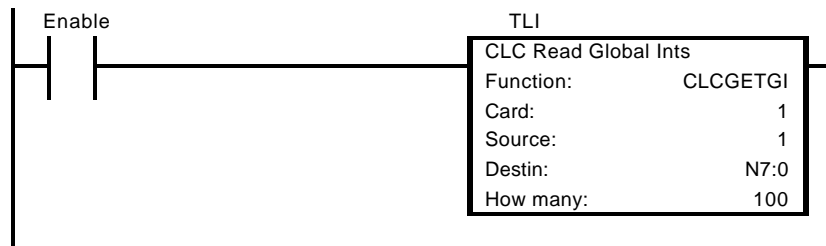
There may be no spaces on either side of = signs. The CARD#, where # is the card number, matches the TLI's "Card" parameter. It needs to be in the range of 0-15. It does not need to agree with Indramat's notion of card number as defined on page 2-8's notion of Card Number. For use with this TLM, "Card Number" is intended to give you the ability to relocate a card to a different memory address without having to change your ladder software. Instead, you simply change your MODULE.LST command line option(s). Specifically, you would change the segment address associated with a Card Number, but not change the Card Number that your ladder logic depends on.

3. The Indramat CLC TLI's

NOTE: Each of these instructions has a maximum "How many" parameter of 100. If the value is more than this, only the first 100 are copied. In order to move more than 100, you must program more than one such instruction.

3.1 CLCGETGI

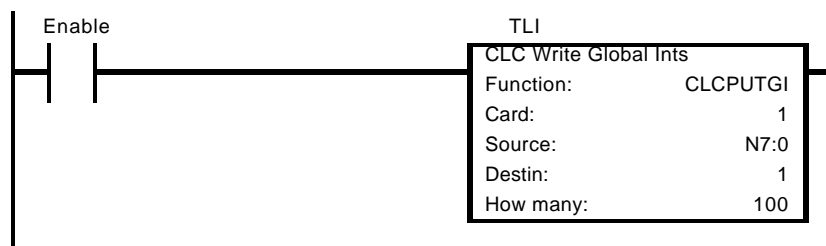
This TLI reads a number of "CLC Global Integers" into SoftPLC's datatable.



Card: Which CLC card number to address.
Source: Which CLC Global Integer number to start copying from.
Destin: Which SoftPLC word address to start copying to. This may be an integer (N) or float (F) section. If integer, then the 32 bit CLC integers are truncated to the least significant 16 bits as they are copied. If float, the 32 bit CLC integers are numerically converted to 32 bit SoftPLC floats, such that the values are preserved as well as possible.
How many: How many registers/words to copy. Maximum of 100 per instruction.

3.2 CLCPUTGI

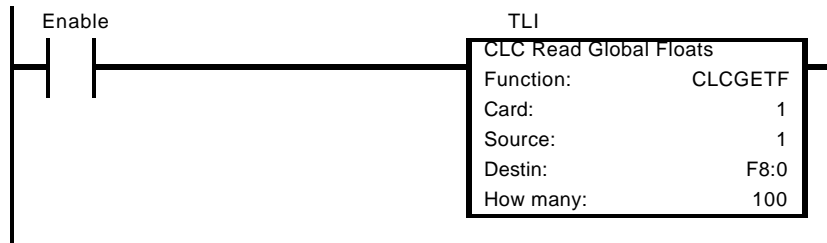
This TLI writes a number of "CLC Global Integers" from SoftPLC's datatable.



Card: Which CLC card number to address.
Source: Which SoftPLC word address to start copying from. This may be an integer (N) or float (F) section. If integer, then the 16 bit SoftPLC integers are signed extended to 32 bit CLC integers as they are copied. If float, the 32 bit SoftPLC float values are converted to CLC integers, such that the values are preserved as well as possible, with fractional components truncated.
Destin: Which CLC register number to start copying to.
How many: How many registers/words to copy. Maximum of 100 per instruction.

3.3 CLCGETGF

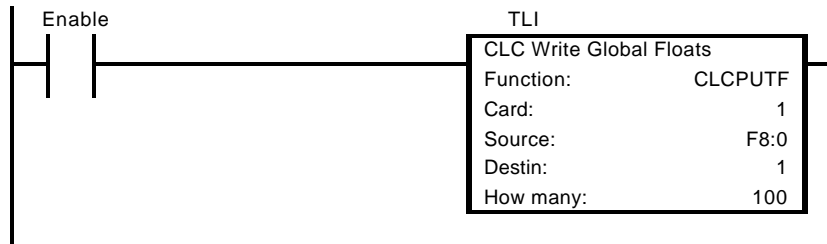
This TLI reads a number of "CLC Global Floating Point Variables" into SoftPLC's datatable.



Card: Which CLC card number to address.
Source: Which CLC float variable number to start copying from.
Destin: Which SoftPLC floating point address to start copying to.
How many: How many floats to copy. Maximum of 100 per instruction.

3.4 CLCPUTGF

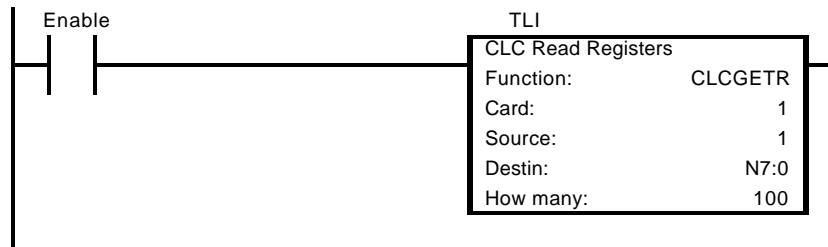
This TLI writes a number of "CLC Global Floating Point Variables" from SoftPLC's datatable.



Card: Which CLC card number to address.
Source: Which SoftPLC floating point address to start copying from.
Destin: Which CLC float variable number to start copying to.
How many: How many floats to copy. Maximum of 100 per instruction.

3.5 CLCGETR

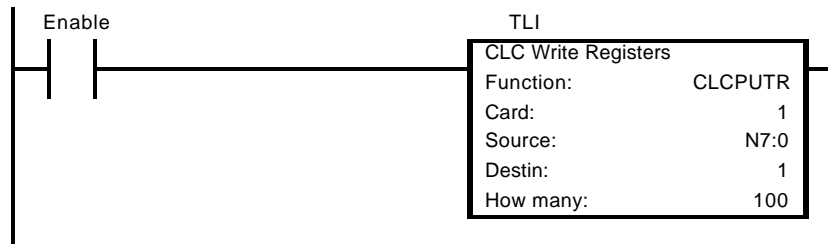
This TLI reads a number of "CLC Global Registers" into SoftPLC's datatable.



Card: Which CLC card number to address.
Source: Which CLC register number to start copying from.
Destin: Which SoftPLC integer word address to start copying to.
How many: How many registers/words to copy. Maximum of 100 per instruction.

3.6 CLCPUTR

This TLI writes a number of "CLC Global Registers" from SoftPLC's datatable.



Card: Which CLC card number to address.
Source: Which SoftPLC integer word address to start copying from.
Destin: Which CLC register number to start copying to.
How many: How many registers/words to copy. Maximum of 100 per instruction.

The MODULE.LST command line would identify all the CLC cards and their memory addresses. This way the TLM would know the specific meaning of the "Card:" parameter for each TLI.

There are consciously no "bit set" commands. The problem with bit set commands is that SoftPLC cannot set or clear a bit as an atomic operation in the dual port memory because the CLC card could also be writing to that register at the same time. (A bit set or clear requires both a read and a write and in between these two atomic operations the CLC could change the original value.) Therefore, the designer will need to designate by his/her own convention which registers and floats are writable by either SoftPLC or the CLC. Any particular register should be writable by only one CPU (readable by either). This way any necessary bit setting operations can be done on the outbound SoftPLC word, i.e. a normal PLC word, that will then subsequently be written to the CLC by way of the CLCPUTGI instruction.