3150-MCM
Example Ladder Logic

Revision 2.1

February 23, 2000

SLC Examples
Application Manual

ProSoft Technology, Inc.
9801 Camino Media, Suite 105
Bakersfield, CA 93311
(661) 664-7208
(661) 664-7233 (fax)
E-mail address: prosoft@prosoft-technology.com
Web Site : http://www.prosoft-technology.com

Quick Start Implementation Guide

Integration of the MCM module into an SLC application is easier if a series of steps are followed. In order to assist the first time users of our products in getting operational quickly, we have come up with this step-by-step implementation guide.

First Time Users

Although the following steps are to assist you in implementing the module, we recommend that you attempt to experiment with the example logic provided on disk with the module or available off our FTP site before laying out your application. This step will allow you to gain insight into how the module works prior to making decisions which will impact the long term success of the installation.

Starting with one of the ladder logic programs provided on disk with the MCM complete the following steps:
If hand entering the ladder logic by hand for the SLC, remember the following:

- Configure the slot as a 1746-BAS module in 5/02 mode
- Be sure to enter the Transfer Enable and Done bits as shown in the example logic

a) Edit the ladder logic provided on disk as needed for the application (See Section 3.0)
Verify rack and slot location in program
Modify ladder instruction addresses as needed

c) Setup the Communication Configuration parameters (See Section 4.2)
Determine each port's communication configuration requirements:
Master or Slave, Parity, Stop Bits, Baud Rate, RTS delay requirements
Identify memory mapping requirements
Set the Read Data, Write Data , and the Command Block Count parameters
Set the Slave and Master Error Table pointers are needed for the application

d) Setup the Command List if configuring a Master (See Section 4.4)
Be sure to review register map of slave device to build most effective memory map

e) Identify the module jumper requirements (See Appendix D)

f) Make up the communication cables (See Section 8). Make sure that no matter what type of connection is being made up that a jumper is in place to satisfy the CTS signal. Normally this signal will be jumpered to RTS.

g) Place processor into the run mode

h) Monitor the data table for the Master and Slave Error Status values (See Section 5.1)

'ProSoft Tested' Test Documents

Through the efforts of our 'ProSoft Tested' Program, we maintain a growing list of devices which we know have been interfaced to our module. In addition, we also have documented several of the devices which we have tested. To access this information, please visit our web site as follows:

http://www.prosoft-technology.com
Select 'Web Site Index'
Select 'MCM Connectivity Listing'
Select 'Test Document' for desired product

Revision Notes

2/23/00	Fix typo error in Expanded Slave register map

SLC Ladder Logic Examples

The following example logic has been provided to assist you in developing applications more effectively.

Slave Mode Examples

Example #1 : Slave Mode w/ Pass-Thru - Minimum Configuration
MCM3EX1S SLC 5/03

Example #2 : Slave Mode w/ Pass-Thru - Expanded Application
MCM3EX2S SLC 5/03

Master Mode Examples

Example #1 : Master Mode - Basic Application
MCM3EX1M SLC 5/03

Example #2 : Master Mode w/ Command Control
MCM3EX2M SLC 5/03

Testing Tools and Suggestions

There are several tools available for assisting in testing the MCM and the associated ladder logic.

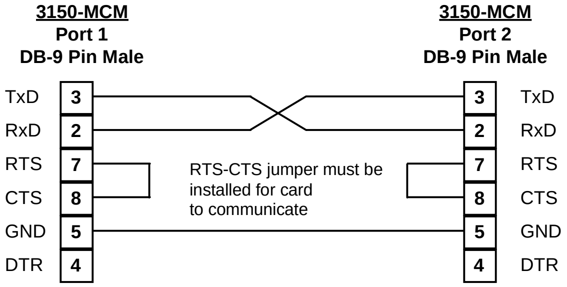
Slave Mode Testing

The simplest test tool we have found for testing out a slave implementation of the MCM product is a Windows based application available off the Internet. We have provided the shareware version of the program on the sample logic diskette under the ‘utils\modscan’ subdirectory. Simply copy this file to your hard drive and ‘Run’ the program from Windows. Instructions are available through the Help File and purchasing instructions are also available.

Master Mode Testing

Testing a Master implementation of the MCM is easily accomplished if the default configuration provided in the example ladder logic is followed. The default configuration places Port 1 as a Master port and Port 2 as a Slave port. In this configuration, the Command List which has been entered in the data table will execute and transfer data between the ports. This method of testing can often be useful when the slave device is not available for testing.

The only external tool necessary to allow Port 1 talk to Port 2 is a short cable with the following configuration:



Slave Mode Example #1 : Slave Mode w/ Pass-Thru - Minimum Configuration

Assumptions

- 10 words of Output Binary data
- 10 words of Input Memory data
- 30 words of Holding Register Data

Port Configuration

PLC Addr	Value	
N[]:7 N[]:17	0	Input Data Start Address - Function Codes 2 and 4 This configuration value determines the beginning address in the module from which the host will begin reading when using Function Codes 2 and 4.
N[]:8 N[]:18	10	Output Data Start Address - Function Codes 1, 5, 15 This configuration value determines the beginnng address in the module from which the host will begin reading and writing data when using Function Codes 1, 5 and 15. Note that in the pass-thru mode the address which the host is writing to will be offset by the value entered here
N[]:9 N[]:19	20	Holding Register Data Start Addr - Function Codes 3, 6, 16 Determines the beginning address in the module in which the host will begin reading and writing data when using Function Codes 3, 6 and 16. Note that when a write command is received in the Pass-Thru mode from a host, the value entered here will be added to the address being received from the host.

System Configuration

N[]:20	1	Read Block Count As a minimum in a slave application we would like to bring back one block which will contain the Slave Error Table (a set of counters and status registers indicating the port status). This is a 20 word block which we will locate at register 50 in our example.
N[]:21	1	Write Block Count This value reflects the number of 50 words blocks that need to be moved to the module to provide data for the host to read. In our example application below we have assumed that the host is reading less than 50 words.
N[]:22	0	Command Block Count When configuring the module in the slave mode only, this value may be set to 0.
N[]:23	50	Slave Error Table Pointer Location Slave Error Table in Module's memory space.
N[]:24	500	Master Error Table Pointer Not used in Slave only configuration, therefore set out of the way (< 3880)
N[]:27	1	Read Block ID Start Value This value determines the starting BTR Block ID number which will be returned from the module. In this example, we want to return only block #1, therefore by setting the value to 1, the module will begin returning from Block #1. The number of blocks returned is determined by the configuration value selected above in the *Read Block Count.*
N[]:28	0	Write Block ID Start Value This value determines the starting BTW Block ID number which be generated by the module. In this example, we wish to write data into Block #0, therefore we will set this value to 0. If we desired to write the data into the module starting at Module Address 100, we would set this configuration value to 2.

Slave Mode example #1 : Slave Mode w/ Pass-Thru - Minimum Configuration
Modbus Memory map

PLC Data		**FC 2**	**FC 4**	**FC 1,5,15**	**FC 3,6,16**
Address	**Module**	**Input Bit**	**Input Register**	**Output Bit**	**Holding Register**
N10	**Address**	**Addresses**	**Addresses**	**Address**	**Address**
0	0	10001 -10016	30001		
1	1	10017 -10032	30002		
2	2	10033 -10048	30003		
3	3	10049 -10064	30004		
4	4	10065 -10080	30005		
5	5	10081 -10096	30006		
6	6	10097 -10112	30007		
7	7	10113 -10128	30008		
8	8	10129 -10144	30009		
9	9	10145 -10160	30010		
10	10			1 - 16	
11	11			17 - 32	
12	12			33 - 48	
13	13			49 - 64	
14	14			65 - 80	
15	15			81 - 96	
16	16			97 - 112	
17	17			113 - 128	
18	18			129 - 144	
19	19			145 - 160	
20	20				40001
21	21				40002
22	22				40003
23	23				40004
24	24				40005
25	25				40006
26	26				40007
27	27				40008
28	28				40009
29	29				40010
30	30				40011
31	31				40012
32	32				40013
33	33				40014
34	34				40015
35	35				40016
36	36				40017
37	37				40018
38	38				40019
39	39				40020
40	40				40021
41	41				40022
42	42				40023
43	43				40024
44	44				40025
45	45				40026
46	46				40027
47	47				40028
48	48				40029
49	49				40030
50	50 to 69	Slave Error Table			

Slave Mode example #1 : Slave Mode w/ Pass-Thru - Minimum Configuration

Data Table File N7

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	
N7:0	9	1	5	0	0	0	0	0	10	20	**Port 1 Config**
N7:10	9	1	5	0	0	0	0	0	10	20	Port 2 Config
N7:20	1	1	0	50	500	0	0	1	0	0	System Config
N7:30	0	0	0	0	0	0	0	0	0	0	Route Table

Data Table File N10

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	
N10:0	1	2	5	4	5	6	7	8	9	10	Write Data To Module
N10:1	11	12	13	14	15	16	17	18	19	20	
N10:20	0	0	0	0	0	0	0	0	0	0	
N10:30	0	0	0	0	0	0	0	0	0	0	
N10:40	0	0	0	0	0	0	0	0	0	0	
N10:50	0	0	0	0	0	0	0	0	0	0	Read Data From Module
N10:60	MC	M	2.	00	11	32	0	0	0	0	- Slave Err Table
N10:70	0	0	0	0	0	0	0	0	0	0	(N10:50-N10:69)
N10:80	0	0	0	0	0	0	0	0	0	0	
N10:90	0	0	0	0	0	0	0	0	0	0	

Data Table File B11

Address	Data (Radix=BINARY)	Address	Data (Radix=BINARY)
B11:0	0000 0000 0000 0000	B11:11	0000 0000 0000 0000
B11:1	0000 0000 0000 0000	B11:12	0000 0000 0000 0000
B11:2	0000 0000 0000 0000	B11:13	0000 0000 0000 0000
B11:3	0000 0000 0000 0000	B11:14	0000 0000 0000 0000
B11:4	0000 0000 0000 0000	B11:15	0000 0000 0000 0000
B11:5	0000 0000 0000 0000	B11:16	0000 0000 0000 0000
B11:6	0000 0000 0000 0000	B11:17	0000 0000 0000 0000
B11:7	0000 0000 0000 0000	B11:18	0000 0000 0000 0000
B11:8	0000 0000 0000 0000	B11:19	0000 0000 0000 0000
B11:9	0000 0000 0000 0000	B11:20	0000 0000 0000 0000
B11:10	0000 0000 0000 0000		

B11:10-19 are used to accept FC 5 bit set/reset commands from the host. The ladder logic takes care in Rung 3:1 of moving the 10 word block back into the module. NOTE that this block location and length are user defined for the application and can easily be modified.

SLC Example Logic

Slave Mode example #1 : Slave Mode w/ Pass-Thru - Minimum Configuration

```
3150-MCM Example #1 Slave Logic
Program Listing                     Processor File: MCM3EX1S.ACH                            Rung 3:0
Rung 3:0
READ DATA FROM MODULE AND DECODE
If the BT Read Block ID is 1, then transfer the module's registers 50-99 into
the data table starting at N10:50.  To add additional data blocks, add new
branches of decode logic.

| TRANSFER  |TRANSFER                                       TEST IF              PASS-THRU         |
| ENABLE    |DONE                                           PASS-THRU            MODE              |
| (required)|(required)                                     COMMAND              HANDLER           |
|    I:1        O:1                                    +GEQ---------------+ +JSR---------------+   |
|----] [--------]/[----------------------------------+-+GRTR THAN OR EQUAL+-+JUMP TO SUBROUTINE+-+-|
|       0          0                                 | |Source A    M1:1.0| |SBR file number  4| | |
|                                                    | |                 *| +------------------+ | |
|               Test if the data received            | |Source B       256|                      | |
|               in the BTR buffer is a pass          | |                  |                      | |
|               through command and if so then       | +------------------+                      | |
|               call SBR 4                           |      DECODE               READ DATA       | |
|                                                    |      BT READ              FROM            | |
|               Transfer the data registers 50 to 99 |      BLOCK ID             MODULE          | |
|               (50 words) to the N10 data file      | +EQU---------------+ +COP---------------+ | |
|               whenever the BTR block ID value is   +-+EQUAL             +-+COPY FILE         +-+ |
|               a 1.  This block is brought into     | |Source A    M1:1.0| |Source     #M1:1.2| | |
|               the ladder logic mainly to be able   | |                 *| |Dest       #N10:50| | |
|               to see the Slave Error Status Table. | |Source B         1| |Length          50| | |
|                                                    | |                  | +------------------+ | |
|                                                    | +------------------+                      | |
|                                                    |                           ENCODES         | |
|                                                    |                           BT WRITE        | |
|               Transfer the BTW Block ID value      |                           BLOCK ID        | |
|               from the read buffer (word 1)        |                      +MOV---------------+ | |
|               into the write buffer (word 0)       +----------------------+MOVE              +-+ |
|               to setup the BTW cycle               |                      |Source      M1:1.1| | |
|                                                    |                      |                 *| | |
|                                                    |                      |Dest        M0:1.0| | |
|                                                    |                      |                 *| | |
|                                                    |                      +------------------+ | |
|               Test if the User wants to            | USER CFG                  ENCODES         | |
|               re-configure the module, and if      | DOWNLOAD                  BT WRITE        | |
|               so then put a 255 into the           | SELECT                    BLOCK ID        | |
|               BTW Block ID position                |      B3              +MOV---------------+ | |
|                                                    +----] [---------------+MOVE              +-+ |
|                                                            0              |Source         255|   |
|                                                                           |                  |   |
|                                                                           |Dest        M0:1.0|   |
|                                                                           |                 *|   |
|                                                                           +------------------+   |
```

SLC Example Logic

Slave Mode example #1 : Slave Mode w/ Pass-Thru - Minimum Configuration

```
Rung 3:1
WRITE DATA TO MODULE
This logic writes the data out to the module for registers 0 to 49.  Also, if
the module is to be configured, the 255 branch will transfer the configuration
block.
NOTE: The relative positioning of the OTE branch is important.  Please duplicate.

| TRANSFER                 TRANSFER          DECODES                                               |
| ENABLE                   DONE              BT WRITE                                              |
| (required)               (required)        BLOCK ID                                              |
|    I:1                      O:1       +EQU---------------+   +COP---------------+                |
|----] [-----------------+----]/[-----+-+EQUAL             +-+-+COPY FILE         +-+-+----------+-|
|       0                |       0    | |Source A    M0:1.0| | |Source     #B11:10| | |          | |
|                        |            | |                 *| | |Dest       #N10:10| | |          | |
|  Move the data for     |            | |Source B         0| | |Length          10| | |          | |
|  module addresses 0 to |            | |                  | | +------------------+ | |          | |
|  49.  In this application           | +------------------+ |                      | |          | |
|  this is data that the host         |                      |     TRANSFER         | |          | |
|  will be reading from the slave     |                      |     TO MODULE        | |          | |
|  (the module)          |            |                      | +COP---------------+ | |          | |
|                        |            |                      +-+COPY FILE         +-+ |          | |
|                        |            |                        |Source      #N10:0|   |          | |
|                        |            |                        |Dest       #M0:1.1|   |          | |
|                        |            |                        |Length          50|   |          | |
|                        |            |                        +------------------+   |          | |
|                        |            |     DECODES                TRANSFER           |          | |
|                        |            |     BT WRITE               TO MODULE          |          | |
|  Configuration of the module        |     BLOCK ID                                  |          | |
|                        |            | +EQU---------------+   +COP---------------+   |          | |
|                        |            +-+EQUAL             +-+-+COPY FILE         +-+-+          | |
|                        |              |Source A    M0:1.0| | |Source       #N7:0| |            | |
|                        |              |                 *| | |Dest       #M0:1.1| |            | |
|                        |              |Source B       255| | |Length          40| |            | |
|                        |              |                  | | +------------------+ |            | |
|                        |              +------------------+ |                      |            | |
|                        |                                   | USER CFG             |            | |
|                        |                                   | DOWNLOAD             |            | |
|                        |                                   | SELECT               |            | |
|                        |                                   |      B3              |            | |
|                        |                                   +----(U)---------------+            | |
|                        |                                           0                           | |
|                        |                                                            TRANSFER   | |
|                        |                                                            DONE       | |
|                        |                                                            (required) | |
|                        |                                                               O:1     | |
|                        +---------------------------------------------------------------( )-----+ |
|                                                                                           0      |

Rung 3:2
|                                                                                                  |
|-----------------------------------------------+END+----------------------------------------------|
|                                                                                                  |
```

SLC Example Logic

Slave Mode example #1 : Slave Mode w/ Pass-Thru - Minimum Configuration

```
3150-MCM Rev 2 Example Ladder Logic
Program Listing                     Processor File: MCM3EX1S.ACH                            Rung 4:0

Rung 4:0
|      DECODE                                                                    ACCEPT            |
|      BT READ                                                                   FC 6              |
|      BLOCK ID                                                                  WRITE             |
| +EQU---------------+                                                      +MOV---------------+   |
|-+EQUAL             +-------------------------------+----------------------+MOVE              +-+-|
| |Source A    M1:1.0|                               |                      |Source      M1:1.4| | |
| |                 *|                               |                      |                 *| | |
| |Source B       256|                               |                      |Dest  N10:[M1:1.3]| | |
| |                  |                               |                      |                 1| | |
| +------------------+                               |                      +------------------+ | |
|               The first branch will handle the     |      DECODE               ACCEPT          | |
|               FC 6 writes from a host.  FC 16      |      ADDRESS              FC 16           | |
|               commands require more work in an     |                           WRITE           | |
|               SLC application. To implement FC 16  | +EQU---------------+ +COP---------------+ | |
|               commands, the programmer must know   +-+EQUAL             +-+COPY FILE         +-+ |
|               the register address and length of the |Source A    M1:1.3| |Source     #M1:1.4|   |
|               write, then must setup a branch of     |                 *| |Dest       #N10:21|   |
|               logic with an EQU test and a COP       |Source B        21| |Length          10|   |
|               programmed to the needed length        |                  | +------------------+   |
|                                                      +------------------+                        |

Rung 4:1
|      DECODE                                                                    COPY ADDR         |
|      BT READ                                                                   AND CMD           |
|      BLOCK ID                                                                  TO WORK           |
|                                                                                REGS              |
| +EQU---------------+                                                      +COP---------------+   |
|-+EQUAL             +-----------------------------------------+------------+COPY FILE         +-+-|
| |Source A    M1:1.0|                                         |            |Source     #M1:1.2| | |
| |                 *|                                         |            |Dest        #N7:52| | |
| |Source B       258|                                         |            |Length           2| | |
| |                  |                                         |            +------------------+ | |
| +------------------+                                         |                                 | |
|                                                              | DECODE     EXECUTE              | |
|               FC 5 handler logic.  The first branch          | FC 5 BIT   SET/RESET            | |
|               copies the write data to an intermediate       | SET/RESET                       | |
|               work register, and the following two branches  |   N7:53       B11               | |
|               take care of latching/unlatching the addressed +----] [--------(L)---------------+ |
|               bit as commanded                               |       0       [N7:52]           | |
|                                                              | DECODE     EXECUTE              | |
|                                                              | FC 5 BIT   SET/RESET            | |
|                                                              | SET/RESET                       | |
|                                                              |   N7:53       B11               | |
|                                                              +----]/[--------(U)---------------+ |
|                                                                      0       [N7:52]             |

Rung 4:2
|                                                                                                  |
|-----------------------------------------------+END+----------------------------------------------|
|
```

Slave Mode Example #2 : Slave Mode w/ Pass-Thru Expanded Application

Assumptions

- 30 words of Output Binary data (N10:40 t0 N10:69)
- 40 words of Input Memory data (N10:0 to N10:39)
- 130 words of Holding Register Data (N10:70 to N10:199)

Port Configuration

PLC Addr	Value	
N[]:7 N[]:17	0	Input Data Start Address - Function Codes 2 and 4 This configuration value determines the beginning address in the module from which the host will begin reading when using Function Codes 2 and 4.
N[]:8 N[]:18	40	Output Data Start Address - Function Codes 1, 5, 15 This configuration value determines the beginnng address in the module from which the host will begin reading and writing data when using Function Codes 1, 5 and 15. Note that in the pass-thru mode the address which the host is writing to will be offset by the value entered here
N[]:9 N[]:19	70	Holding Register Data Start Addr - Function Codes 3, 6, 16 Determines the beginning address in the module in which the host will begin reading and writing data when using Function Codes 3, 6 and 16. Note that when a write command is received in the Pass-Thru mode from a host, the value entered here will be added to the address being received from the host.

System Configuration

N[]:20	1	Read Block Count As a minimum in a slave application we would like to bring back one block which will contain the Slave Error Table (a set of counters and status registers indicating the port status). This is a 20 word block which we will locate at register 200 in our example.
N[]:21	4	Write Block Count This value reflects the number of 50 words blocks that need to be moved to the module to provide data for the host to read. In our example application below we have assumed that the host is reading 200 words (4 blocks)
N[]:22	0	Command Block Count When configuring the module in the slave mode only, this value may be set to 0.
N[]:23	200	Slave Error Table Pointer Location Slave Error Table in Module's memory space.
N[]:24	500	Master Error Table Pointer Not used in Slave only configuration, therefore set out of the way (< 3880)
N[]:27	4	Read Block ID Start Value This value determines the starting BTR Block ID number which will be returned from the module. In this example, we want to return only block #4, therefore by setting the value to 4, the module will begin returning from Block #4. The number of blocks returned is determined by the configuration value selected above in the *Read Block Count.*
N[]:28	0	Write Block ID Start Value This value determines the starting BTW Block ID number which be generated by the module. In this example, we wish to write data into Block #0, therefore we will set this value to 0. If we desired to write the data into the module starting at Module Address 100, we would set this configuration value to 2.

Slave Mode Example #2 : Slave Mode w/ Pass-Thru - Expanded Configuration
Modbus Memory map

PLC Data		**FC 2**	**FC 4**	**FC 1,5,15**	**FC 3,6,16**
Address	**Module**	**Input Bit**	**Input Register**	**Output Bit**	**Holding Register**
N10	**Address**	**Addresses**	**Addresses**	**Address**	**Address**
0	0	10001 -10016	30001		
1	1	10017 -10032	30002		
2	2	10033 -10048	30003		
3	3	10049 -10064	30004		
4	4	10065 -10080	30005		
up to	up to				
38	38	10609 -10624	30039		
39	39	10625 -10640	30040		
40	40			1 - 16	
41	41			17 - 32	
42	42			33 - 48	
43	43			49 - 64	
44	44			65 - 80	
up to	up to				
68	68			449 - 464	
69	69			465 - 480	
70	70				40001
71	71				40002
72	72				40003
73	73				40004
74	74				40005
75	75				40006
76	76				40007
77	77				40008
78	78				40009
79	79				40010
80	80				40011
81	81				40012
189	189				40120
190	190				40121
191	191				40122
192	192				40123
193	193				40124
194	194				40125
195	195				40126
196	196				40127
197	197				40128
198	198				40129
199	199				40130
200	200 to 219	Slave Error Table			

Slave Mode Example #2 : Slave Mode w/ Pass-Thru - Expanded Configuration

Data Table File N7

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	
N7:0	9	1	5	0	0	0	0	0	40	70	**Port 1 Config**
N7:10	9	1	5	0	0	0	0	0	40	70	Port 2 Config
N7:20	1	4	0	200	500	0	0	4	0	0	System Config
N7:30	0	0	0	0	0	0	0	0	0	0	Route Table

Data Table File N10

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	
N10:0	1	2	5	4	5	6	7	8	9	10	Write Data To Module
N10:10	11	12	13	14	15	16	17	18	19	20	
N10:20	0	0	0	0	0	0	0	0	0	0	
N10:30	0	0	0	0	0	0	0	0	0	0	
N10:40	0	0	0	0	0	0	0	0	0	0	
N10:50	0	0	0	0	0	0	0	0	0	0	
N10:60	0	0	0	0	0	0	0	0	0	0	
N10:70	0	0	0	0	0	0	0	0	0	0	
N10:80	0	0	0	0	0	0	0	0	0	0	
N10:90	0	0	0	0	0	0	0	0	0	0	
N10:100	0	0	0	0	0	0	0	0	0	0	
N10:110	0	0	0	0	0	0	0	0	0	0	
N10:120	0	0	0	0	0	0	0	0	0	0	
N10:130	0	0	0	0	0	0	0	0	0	0	
N10:140	0	0	0	0	0	0	0	0	0	0	
N10:150	0	0	0	0	0	0	0	0	0	0	
N10:160	0	0	0	0	0	0	0	0	0	0	
N10:170	0	0	0	0	0	0	0	0	0	0	
N10:180	0	0	0	0	0	0	0	0	0	0	
N10:190	0	0	0	0	0	0	0	0	0	0	
N10:200	0	0	0	0	0	0	0	0	0	0	Read Data From Module
N10:210	MC	M	2.	00	11	32	0	0	0	0	-Slave Err Table
N10:220	0	0	0	0	0	0	0	0	0	0	(N10:200-N10:219)
N10:230	0	0	0	0	0	0	0	0	0	0	
N10:240	0	0	0	0	0	0	0	0	0	0	

Data Table File B11

Address	Data (Radix=BINARY)	Address	Data (Radix=BINARY)
B11:40	0000 0000 0000 0000	B11:51	0000 0000 0000 0000
B11:41	0000 0000 0000 0000	B11:52	0000 0000 0000 0000
B11:42	0000 0000 0000 0000	B11:53	0000 0000 0000 0000
B11:43	0000 0000 0000 0000	B11:54	0000 0000 0000 0000
B11:44	0000 0000 0000 0000	B11:55	0000 0000 0000 0000
B11:45	0000 0000 0000 0000	B11:56	0000 0000 0000 0000
B11:46	0000 0000 0000 0000	B11:57	0000 0000 0000 0000
B11:47	0000 0000 0000 0000	B11:58	0000 0000 0000 0000
B11:48	0000 0000 0000 0000	B11:59	0000 0000 0000 0000
B11:49	0000 0000 0000 0000	up to	
B11:50	0000 0000 0000 0000	B11:69	0000 0000 0000 0000

B11:40-69 are used to accept FC 5 bit set/reset commands from the host. The ladder logic takes care in Rung 3:1 of moving the 30 word block back into the module. NOTE that this block location and length are user defined for the application and can easily be modified.

SLC Example Logic
Slave Mode Example #2 : Slave Mode w/ Pass-Thru - Expanded Configuration

```
3150-MCM Master Example #2                                                                January 03, 1997   Page 2
Program Listing                            Processor File: MCM3EX2S.ACH                                    Rung 3:0

Rung 3:0
READ DATA FROM MODULE AND DECODE
If the BT Read Block ID is 1, then transfer the module's registers 50-99 into
the data table starting at N10:50.  To add additional data blocks, add new
branches of decode logic.
| TRANSFER  |TRANSFER                                                      TEST IF              PASS-THRU         |
| ENABLE    |DONE                                                          PASS THRU            MODE              |
| (required)|(required)                                                    COMMAND              HANDLER           |
|    I:1        O:1                                                   +GEQ---------------+ +JSR---------------+   |
|----] [--------]/[-------------------------------------------------+-+GRTR THAN OR EQUAL+-+JUMP TO SUBROUTINE+-+-|
|       0          0                                                | |Source A    M1:1.0| |SBR file number  4| | |
|                                                                   | |                 *| +------------------+ | |
|               Test if the data received in the M1                 | |Source B       256|                      | |
|               file from the module is a pass-through              | |                  |                      | |
|               command and if so then call SBR 4                   | +------------------+                      | |
|                                                                   |      DECODE               READ DATA       | |
|                                                                   |      BT READ              FROM            | |
|                                                                   |      BLOCK ID             MODULE          | |
|               Transfer the data registers 200 to 249 (50 words)   | +EQU---------------+ +COP---------------+ | |
|               from the module to the N10 file whenever the BTR    +-+EQUAL             +-+COPY FILE         +-+ |
|               Block ID is 4.  This block is brought into the      | |Source A    M1:1.0| |Source     #M1:1.2| | |
|               data table to gain access to the Slave Error Table  | |                 *| |Dest      #N10:200| | |
|                                                                   | |Source B         4| |Length          50| | |
|                                                                   | |                  | +------------------+ | |
|                                                                   | +------------------+                      | |
|                                                                   |                           ENCODES         | |
|                                                                   |                           BT WRITE        | |
|               Transfer the BTW Block ID value from the read       |                           BLOCK ID        | |
|               buffer (word 1) into the write buffer (word 0)      |                      +MOV---------------+ | |
|               to setup the BTW cycle                              +----------------------+MOVE              +-+ |
|                                                                   |                      |Source      M1:1.1| | |
|                                                                   |                      |                 *| | |
|                                                                   |                      |Dest        M0:1.0| | |
|                                                                   |                      |                 *| | |
|                                                                   |                      +------------------+ | |
|                                                                   | USER CFG                  ENCODES         | |
|               Test if the User wants to re-configure              | DOWNLOAD                  BT WRITE        | |
|               the module and if so then put a 255 into the        | SELECT                    BLOCK ID        | |
|               BTW Block ID position                               |      B3              +MOV---------------+ | |
|                                                                   +----] [---------------+MOVE              +-+ |
|                                                                           0              |Source         255|   |
|                                                                                          |                  |   |
|                                                                                          |Dest        M0:1.0|   |
|                                                                                          |                 *|   |
|                                                                                          +------------------+   |
```

SLC Example Logic
Slave Mode Example #2 : Slave Mode w/ Pass-Thru - Expanded Configuration

```
3150-MCM Master Example #2                                                                January 03, 1997   Page 3
Program Listing                            Processor File: MCM3EX2S.ACH                                    Rung 3:1
Rung 3:1
| TRANSFER        TRANSFER                                                           COPY                          |
| ENABLE          DONE                                                               OUTPUT                        |
| (required)      (required)                                                         IMAGE FOR                     |
|                                                                                    MODULE                        |
|    I:1             O:1                                                        +COP---------------+               |
|----] [--------+----]/[-----+--------------------------------------------------+COPY FILE         +-+-----------+-|
|       0       |       0    |                                                  |Source     #B11:40| |           | |
|               |            |                                                  |Dest       #N10:40| |           | |
|    Copy the Output Data Image                                                 |Length          30| |           | |
|    into the N10 Integer as a buffer                                           +------------------+ |           | |
|               |            |      VERIFY BTW             DETERMINE                                 |           | |
|               |            |      ID VALUE               FILE ADDR                                 |           | |
|               |            |      W/IN LIMIT             POINTER                                   |           | |
|               |            | +LES---------------+   +CPT---------------+                           |           | |
|    Setup using indirect    +-+LESS THAN         +-+-+COMPUTE           +------+--------------------+           | |
|    addressing to copy the  | |Source A    M0:1.0| | |Dest         N7:40|      |                    |           | |
|    data from the data table| |                 *| | |                 0|      |                    |           | |
|    into the module. The    | |Source B     N7:21| | |Expression        |      |                    |           | |
|    indirect addressing     | |                 4| | |M0:1.0 * 50       |      |                    |           | |
|    method is convenient when +------------------+ | +------------------+      |                    |           | |
|    a large number of branches                      |        TRANSFER           |                    |           | |
|    would otherwise be required.                    |        TO MODULE          |                    |           | |
|               |            |                      | +COP--------------------+ |                    |           | |
|               |            |                      +-+COPY FILE              +-+                    |           | |
|               |            |                        |Source     #N10:[N7:40]|                      |           | |
|               |            |                        |Dest            #M0:1.1|                      |           | |
|               |            |                        |Length               50|                      |           | |
|               |            |                        +-----------------------+                      |           | |
|               |            |      DECODES                TRANSFER                                  |           | |
|               |            |      BT WRITE               TO MODULE                                 |           | |
|               |            |      BLOCK ID                                                         |           | |
|    Configuration of the    | +EQU---------------+   +COP---------------+                           |           | |
|    module     |            +-+EQUAL             +-+-+COPY FILE         +------+--------------------+           | |
|               |              |Source A    M0:1.0| | |Source       #N7:0|      |                                | |
|               |              |                 *| | |Dest       #M0:1.1|      |                                | |
|               |              |Source B       255| | |Length          30|      |                                | |
|               |              |                  | | +------------------+      |                                | |
|               |              +------------------+ |                           |                                | |
|               |                                   | USER CFG                  |                                | |
|               |                                   | DOWNLOAD                  |                                | |
|               |                                   | SELECT                    |                                | |
|               |                                   |      B3                   |                                | |
|               |                                   +----(U)--------------------+                                | |
|               |                                          0                                                     | |
|               |                                                                                    TRANSFER    | |
|               |                                                                                    DONE        | |
|               |                                                                                    (required)  | |
|               |                                                                                       O:1      | |
|               +---------------------------------------------------------------------------------------( )------+ |
|                                                                                                          0       |

Rung 3:2
|                                                                                                                  |
|-------------------------------------------------------+END+------------------------------------------------------|
|                                                                                                                  |
```

SLC Example Logic

Slave Mode Example #2 : Slave Mode w/ Pass-Thru - Expanded Configuration

```
3150-MCM Rev 2 Example Ladder Logic
Program Listing                     Processor File: MCM3EX1S.ACH                            Rung 4:0

Rung 4:0
|      DECODE                                                                    ACCEPT            |
|      BT READ                                                                   FC 6              |
|      BLOCK ID                                                                  WRITE             |
| +EQU---------------+                                                      +MOV---------------+   |
|-+EQUAL             +-------------------------------+----------------------+MOVE              +-+-|
| |Source A    M1:1.0|                               |                      |Source      M1:1.4| | |
| |                 *|                               |                      |                 *| | |
| |Source B       256|                               |                      |Dest  N10:[M1:1.3]| | |
| |                  |                               |                      |                 1| | |
| +------------------+                               |                      +------------------+ | |
|               The first branch will handle the     |      DECODE               ACCEPT          | |
|               FC 6 writes from a host.  FC 16      |      ADDRESS              FC 16           | |
|               commands require more work in an     |                           WRITE           | |
|               SLC application. To implement FC 16  | +EQU---------------+ +COP---------------+ | |
|               commands, the programmer must know   +-+EQUAL             +-+COPY FILE         +-+ |
|               the register address and length of the |Source A    M1:1.3| |Source     #M1:1.4|   |
|               write, then must setup a branch of     |                 *| |Dest       #N10:21|   |
|               logic with an EQU test and a COP       |Source B        21| |Length         10|   |
|               programmed to the needed length        |                  | +------------------+   |
|                                                      +------------------+                        |

Rung 4:1
|      DECODE                                                                    COPY ADDR         |
|      BT READ                                                                   AND CMD           |
|      BLOCK ID                                                                  TO WORK           |
|                                                                                REGS              |
| +EQU---------------+                                                      +COP---------------+   |
|-+EQUAL             +-----------------------------------------+------------+COPY FILE         +-+-|
| |Source A    M1:1.0|                                         |            |Source     #M1:1.2| | |
| |                 *|                                         |            |Dest        #N7:52| | |
| |Source B       258|                                         |            |Length           2| | |
| |                  |                                         |            +------------------+ | |
| +------------------+                                         |                                 | |
|                                                              | DECODE     EXECUTE              | |
|               FC 5 handler logic.  The first branch          | FC 5 BIT   SET/RESET            | |
|               copies the write data to an intermediate       | SET/RESET                       | |
|               work register, and the following two branches  |   N7:53       B11               | |
|               take care of latching/unlatching the addressed +----] [--------(L)---------------+ |
|               bit as commanded                               |       0       [N7:52]           | |
|                                                              | DECODE     EXECUTE              | |
|                                                              | FC 5 BIT   SET/RESET            | |
|                                                              | SET/RESET                       | |
|                                                              |   N7:53       B11               | |
|                                                              +----]/[--------(U)---------------+ |
|                                                                      0       [N7:52]             |

Rung 4:2
|                                                                                                  |
|-----------------------------------------------+END+----------------------------------------------|
|
```

Master Mode Example #1 : Master Mode - Basic Application

The following example provides an example of the MCM module in a Master application. In this example we have setup Port 1 as a Master. Port 2 has been setup as a Slave for testing purposes only, but you may program it as needed. In order to test the logic which we have provided, install a looped cable from Port 1 to Port 2 as shown in the beginning of this manual.

Assumptions

- Read 200 words from Module (values from slaves and Master Error Table)
- Write 50 words to module (for writing to slaves)

System Configuration

N[]:20	4	Read Block Count	This value represents the total number of 50 word data blocks that we want to read back from the module into the PLC/SLC data table. In this application we have setup to read back registers 0 to 199.
N[]:21	1	Write Block Count	This value reflects the number of 50 words blocks that need to be moved to the module to provide data for the module to write to the slaves.
N[]:22	2	Command Block Count	This value represents the number of Command Blocks (5 commands per block) that we would like to send to the module. In this application we wanted to allow for 10 commands, even if we have only configured 5 of them
N[]:23	130	Slave Table Ptr	Location Slave Error Table in Module's memory space. In a Master application we still would like to bring back this table in order to have the module firmware revision information
N[]:24	150	Master Table Ptr	Location of the Master Error Table in the Module's memory space. In this application we have located this table after the Slave Error Table. Since we do not have more than 50 commands we are not concerned about the remainder of the table. In fact since we have allowed for only 10 commands, we could have set the Master Table Pointer to 189 if we had wanted to conserve memory
N[]:27	0	Read Block ID Start Value	This value determines the starting BTR Block ID number which will be returned from the module.
N[]:28	4	Write Block ID Start Value	This value determines the starting BTW Block ID number which be generated by the module. In this example, we wish to write data into Block #4, therefore we will set this value to 4.

Master Mode example #1 : Master Mode - Basic Application

Data Table File N7:0

Address	0	1	2	3	4	5	6	7	8	9	
N7:0	0	0	5	0	0	0	0	0	0	0	Port 1 Config
N7:10	1	1	5	0	0	0	0	0	0	0	Port 2 Config
N7:20	4	1	2	130	150	0	0	0	4	0	System Config
N7:30	0	0	0	0	0	0	0	0	0	0	Route Table
N7:40	0	0	0	0	0	0	0	0	0	0	
											Command List
N7:50	1	1	3	200	10	0	0	0	0	0	Command #1
N7:60	1	1	3	210	10	10	0	0	0	0	Command #2
N7:70	1	1	4	220	10	20	0	0	0	0	Command #3
N7:80	1	1	3	200	10	30	0	0	0	0	Command #4
N7:90	1	1	16	200	40	40	0	0	0	0	Command #5
N7:100	0	0	0	0	0	0	0	0	0	0	Command #6
N7:110	0	0	0	0	0	0	0	0	0	0	Command #7
N7:120	0	0	0	0	0	0	0	0	0	0	Command #8
N7:130	0	0	0	0	0	0	0	0	0	0	Command #9
N7:140	0	0	0	0	0	0	0	0	0	0	Command #10

Data Table File N10:0

Address	0	1	2	3	4	5	6	7	8	9	
N10:0	200	201	202	203	0	0	0	0	0	0	Read Data Block
N10:10	210	211	212	213	0	0	0	0	0	0	from Module
N10:20	220	221	222	223	0	0	0	0	0	0	Reg 0 to 199
N10:30	200	201	202	203	0	0	0	0	0	0	
N10:40	200	201	202	203	0	0	0	0	0	0	
N10:50	210	211	212	213	0	0	0	0	0	0	
N10:60	220	221	222	223	0	0	0	0	0	0	
N10:70	200	201	202	203	0	0	0	0	0	0	
N10:80	0	0	0	0	0	0	0	0	0	0	
N10:90	0	0	0	0	0	0	0	0	0	0	
N10:100	0	0	0	0	0	0	0	0	0	0	
N10:110	0	0	0	0	0	0	0	0	0	0	
N10:120	0	0	0	0	0	0	0	0	0	0	
N10:130	0	0	0	0	0	0	0	932	932	932	Slave Error Table
N10:140	MC	M	2.	00	11	32	0	0	0	0	
N10:150	1	0	0	0	0	0	0	0	0	0	Master Error Table
N10:160	0	0	0	0	0	0	0	0	0	0	
N10:170	0	0	0	0	0	0	0	0	0	0	
N10:180	0	0	0	0	0	0	0	0	0	0	
N10:190	0	0	0	0	0	0	0	0	0	0	
N10:200	200	201	202	203	0	0	0	0	0	0	Write Data Block
N10:210	210	211	212	213	0	0	0	0	0	0	to Module
N10:220	220	221	222	223	0	0	0	0	0	0	Reg 200 to 249
N10:230	0	0	0	0	0	0	0	0	0	0	
N10:240	0	0	0	0	0	0	0	0	0	0	

SLC Ladder Logic

Master Mode example #1 : Master Mode - Basic Application

```
3150-MCM Example Ladder Logic                                                                  January 01, 1997   Page 2
Program Listing                               Processor File: MCM3EX1M.ACH
Rung 3:0
REGISTER TRANSFER FROM MODULE
If the BT Read Block ID is between 0 and 3 inclusive, then transfer the
module's registers into the data table.  To add additional data blocks, simply
add additional decoding logic.
| TRANSFER  |TRANSFER                                                           DECODE               READS             |
| ENABLE    |DONE                                                               BT READ              BLOCK             |
| (required)|(required)                                                         BLOCK ID                               |
|    I:1        O:1                                                         +EQU---------------+ +COP---------------+   |
|----] [--------]/[------------------------------------------------------+-+EQUAL             +-+COPY FILE         +-+-|
|       0          0                                                     | |Source A    M1:1.0| |Source     #M1:1.2| | |
|                                                                        | |                 *| |Dest        #N10:0| | |
|                                                                        | |Source B         0| |Length          50| | |
|                                                                        | |                  | +------------------+ | |
|                                                                        | +------------------+                      | |
|                                                                        |      DECODE               COPY BLOCK      | |
|                                                                        |      BT READ              FROM            | |
|               Decode each of the BTR Block ID numbers which            |      BLOCK ID             MODULE          | |
|               will be returned from the module and copy the            | +EQU---------------+ +COP---------------+ | |
|               data buffer into the data table. An example              +-+EQUAL             +-+COPY FILE         +-+ |
|               using indirect addressing to reduce the branch           | |Source A    M1:1.0| |Source     #M1:1.2| | |
|               count can be viewed in Master Example #2.                | |                 *| |Dest       #N10:50| | |
|                                                                        | |Source B         1| |Length          50| | |
|                                                                        | |                  | +------------------+ | |
|                                                                        | +------------------+                      | |
|                                                                        |      DECODE               COPY BLOCK      | |
|                                                                        |      BT READ              FROM            | |
|                                                                        |      BLOCK ID             MODULE          | |
|                                                                        | +EQU---------------+ +COP---------------+ | |
|                                                                        +-+EQUAL             +-+COPY FILE         +-+ |
|                                                                        | |Source A    M1:1.0| |Source     #M1:1.2| | |
|                                                                        | |                 *| |Dest      #N10:100| | |
|                                                                        | |Source B         2| |Length          50| | |
|                                                                        | |                  | +------------------+ | |
|                                                                        | +------------------+                      | |
|                                                                        |      DECODE               COPY BLOCK      | |
|                                                                        |      BT READ              FROM            | |
|                                                                        |      BLOCK ID             MODULE          | |
|                                                                        | +EQU---------------+ +COP---------------+ | |
|                                                                        +-+EQUAL             +-+COPY FILE         +-+ |
|                                                                        | |Source A    M1:1.0| |Source     #M1:1.2| | |
|                                                                        | |                 *| |Dest      #N10:150| | |
|                                                                        | |Source B         3| |Length          50| | |
|                                                                        | |                  | +------------------+ | |
|                                                                        | +------------------+                      | |
|                                                                        |                           ENCODES         | |
|                                                                        |                           BLOCK           | |
|               Transfer the BTW Block ID value from the                 |                      +MOV---------------+ | |
|               read buffer (word 1) to the write buffer (word 0)        +----------------------+MOVE             +-+ |
|               to setup the BTW cycle                                   |                      |Source      M1:1.1| | |
|                                                                        |                      |                 *| | |
|                                                                        |                      |Dest        M0:1.0| | |
|                                                                        |                      |                 *| | |
|                                                                        |                      +------------------+ | |
|                                                                        | USER CFG                 ENCODES         | |
|                                                                        | DOWNLOAD                 BLOCK           | |
|               Test if the User wants to re-configure the               | SELECT                                   | |
|               module, and if so then put a 255 into the BTW Block      |      B3              +MOV---------------+ | |
|               ID position                                              +----] [---------------+MOVE             +-+ |
|                                                                                0              |Source         255|   |
|                                                                                               |                  |   |
|                                                                                               |Dest        M0:1.0|   |
|                                                                                               |                 *|   |
|                                                                                               +------------------+   |
```

SLC Ladder Logic

Master Mode example #1 : Master Mode - Basic Application

```
3150-MCM Example Ladder Logic                                                                  January 01, 1997   Page 3
Program Listing                               Processor File: MCM3EX1M.ACH                                      Rung 3:1

Rung 3:1
WRITE DATA, COMMAND LIST OR CONFIGURATION BLOCK TO MODULE
Based on the value in the BTW Block ID, either the data or the command list is
moved to the module, or configuration parameters are moved to the module.  To
move additional data, add new branches
| TRANSFER                  TRANSFER          DECODES                                     TRANSFER                     |
| ENABLE                    DONE              BT WRITE                                    TO MODULE                    |
| (required)                (required)        BLOCK ID                                                                 |
|    I:1                       O:1       +EQU---------------+                        +COP---------------+              |
|----] [------------------+----]/[-----+-+EQUAL             +------------------------+COPY FILE         +-+----------+-|
|       0                 |       0    | |Source A    M0:1.0|                        |Source    #N10:200| |          | |
|        Move the data for the module  | |                 *|                        |Dest       #M0:1.1| |          | |
|        addresses 200 to 249.  In a   | |Source B         4|                        |Length          50| |          | |
|        Master application, these     | |                  |                        +------------------+ |          | |
|        register locations will be used +------------------+                                             |          | |
|        to write values to the slaves.|      DECODES                                     TRANSFER        |          | |
|                         |            |      BT WRITE                                    TO MODULE       |          | |
|                         |            |      BLOCK ID                                                    |          | |
|                         |            | +EQU---------------+                        +COP---------------+ |          | |
|        Command List     |            +-+EQUAL             +------------------------+COPY FILE         +-+          | |
|        Block IDs 80 and 81 (through  | |Source A    M0:1.0|                        |Source      #N7:50| |          | |
|        99) are used to transfer the  | |                 *|                        |Dest       #M0:1.1| |          | |
|        Command List to the module.   | |Source B        80|                        |Length          50| |          | |
|        Each ‘block’ moves 5 commands | |                  |                        +------------------+ |          | |
|        to the module.   |            | +------------------+                                             |          | |
|                         |            |      DECODES                                     TRANSFER        |          | |
|                         |            |      BT WRITE                                    TO MODULE       |          | |
|                         |            |      BLOCK ID                                                    |          | |
|                         |            | +EQU---------------+                        +COP---------------+ |          | |
|                         |            +-+EQUAL             +------------------------+COPY FILE         +-+          | |
|                         |            | |Source A    M0:1.0|                        |Source     #N7:100| |          | |
|                         |            | |                 *|                        |Dest       #M0:1.1| |          | |
|                         |            | |Source B        81|                        |Length          50| |          | |
|                         |            | |                  |                        +------------------+ |          | |
|                         |            | +------------------+                                             |          | |
|                         |            |      DECODES                 TRANSFER                            |          | |
|                         |            |      BT WRITE                TO MODULE                           |          | |
|                         |            |      BLOCK ID                                                    |          | |
|                         |            | +EQU---------------+   +COP---------------+                      |          | |
|        Configuration of the module   +-+EQUAL             +-+-+COPY FILE         +-+--------------------+          | |
|                         |               |Source A    M0:1.0| | |Source       #N7:0| |                               | |
|                         |               |                 *| | |Dest       #M0:1.1| |                               | |
|                         |               |Source B       255| | |Length         40| |                               | |
|                         |               |                  | | +------------------+ |                               | |
|                         |               +------------------+ |                      |                               | |
|                         |                                    | USER CFG             |                               | |
|                         |                                    | DOWNLOAD             |                               | |
|                         |                                    | SELECT               |                               | |
|                         |                                    |      B3              |                               | |
|                         |                                    +----(U)---------------+                               | |
|                         |                                            0                                              | |
|                         |                                                                              TRANSFER   | |
|                         |                                                                              DONE       | |
|                         |                                                                              (required) | |
|                         |                                                                                         | |
|                         |                                                                                 O:1     | |
|                         +---------------------------------------------------------------------------------( )-----+ |
|                                                                                                              0      |

Rung 3:2
|                                                                                                                      |
|---------------------------------------------------------+END+--------------------------------------------------------|
|                                                                                                                      |
```

Master Mode Example #2: Master Mode w/ Command Control Enabled

The following example provides an example of the MCM module in a Master application. In this example we have setup Port 1 as a Master. Port 2 has been setup as a Slave for testing purposes only, but you may program it as needed. In order to test the logic which we have provided, install a looped cable from Port 1 to Port 2 as shown in the beginning of this manual.

Assumptions

- Read 200 words from Module (values from slaves and Master Error Table)
- Write 50 words to module (for writing to slaves)

System Configuration

N[]:20	4	Read Block Count	This value represents the total number of 50 word data blocks that we want to read back from the module into the PLC/SLC data table. In this application we have setup to read back registers 0 to 199.
N[]:21	1	Write Block Count	This value reflects the number of 50 words blocks that need to be moved to the module to provide data for the module to write to the slaves.
N[]:22	2	Command Block Count	This value represents the number of Command Blocks (5 commands per block) that we would like to send to the module. In this application we wanted to allow for 10 commands, even if we have only configured 5 of them
N[]:23	130	Slave Table Ptr	Location Slave Error Table in Module's memory space. In a Master application we still would like to bring back this table in order to have the module firmware revision information
N[]:24	150	Master Table Ptr	Location of the Master Error Table in the Module's memory space. In this application we have located this table after the Slave Error Table. Since we do not have more than 50 commands we are not concerned about the remainder of the table. In fact since we have allowed for only 10 commands, we could have set the Master Table Pointer to 189 if we had wanted to conserve memory
N[]:27	0	Read Block ID Start Value	This value determines the starting BTR Block ID number which will be returned from the module.
N[]:28	4	Write Block ID Start Value	This value determines the starting BTW Block ID number which be generated by the module. In this example, we wish to write data into Block #4, therefore we will set this value to 4.

Master Mode Example #2 : Master Mode w/ Command Control Enabled

```
Data Table File N7:0

  Address     0      1      2      3      4      5      6      7      8      9
N7:0           0      0      5      0      0      0      0      0      0      0  Port 1 Config
N7:10          1      1      5      0      0      0      0      0      0      0  Port 2 Config
N7:20          4      1      2    130    150      0      0      0      4      0  System Config
N7:30          0      0      0      0      0      0      0      0      0      0  Route Table
N7:40          0      0      0      0      0      0      0      0      0      0
                                                                                 Command List
N7:50          9      1      3    200     10      0      0      0      0      0  Command #1
N7:60          9      1      3    210     10     10      0      0      0      0  Command #2
N7:70          9      1      4    220     10     20      0      0      0      0  Command #3
N7:80          1      1      3    200     10     30      0      0      0      0  Command #4
N7:90          1      1     16    200     40     40      0      0      0      0  Command #5
N7:100         0      0      0      0      0      0      0      0      0      0  Command #6
N7:110         0      0      0      0      0      0      0      0      0      0  Command #7
N7:120         0      0      0      0      0      0      0      0      0      0  Command #8
N7:130         0      0      0      0      0      0      0      0      0      0  Command #9
N7:140         0      0      0      0      0      0      0      0      0      0  Command #10
```

```
Data Table File B9

Address   Data  (Radix=BINARY)    Address   Data  (Radix=BINARY)          B9 is used for Command Control.
B9:0      0000 0000 0000 0000     B9:11     0000 0000 0000 0000           Words 0 to 5 : Command Enable
B9:1      0000 0000 0000 0000     B9:12     0000 0000 0000 0000           Words 6 to 11: Command Done
B9:2      0000 0000 0000 0000     B9:13     0000 0000 0000 0000           Words 12 to 17:Command Error
B9:3      0000 0000 0000 0000     B9:14     0000 0000 0000 0000
B9:4      0000 0000 0000 0000     B9:15     0000 0000 0000 0000
B9:5      0000 0000 0000 0000     B9:16     0000 0000 0000 0000
B9:6      0000 0000 0000 0000     B9:17     0000 0000 0000 0000
B9:7      0000 0000 0000 0000     B9:18     0000 0000 0000 0000
B9:8      0000 0000 0000 0000
B9:9      0000 0000 0000 0000
B9:10     0000 0000 0000 0000
```

```
Data Table File N10:0

  Address     0      1      2      3      4      5      6      7      8      9
N10:0        200    201    202    203      0      0      0      0      0      0  Read Data Block
N10:10       210    211    212    213      0      0      0      0      0      0  from Module
N10:20       220    221    222    223      0      0      0      0      0      0  Reg 0 to 199
N10:30       200    201    202    203      0      0      0      0      0      0
N10:40       200    201    202    203      0      0      0      0      0      0
N10:50       210    211    212    213      0      0      0      0      0      0
N10:60       220    221    222    223      0      0      0      0      0      0
N10:70       200    201    202    203      0      0      0      0      0      0
N10:80         0      0      0      0      0      0      0      0      0      0
N10:90         0      0      0      0      0      0      0      0      0      0
N10:100        0      0      0      0      0      0      0      0      0      0
N10:110        0      0      0      0      0      0      0      0      0      0
N10:120        0      0      0      0      0      0      0      0      0      0

N10:130        0      0      0      0      0      0      0    932    932    932  Slave Error Table
N10:140       MC      M     2.     00     11     32      0      0      0      0

N10:150        1      0      0      0      0      0      0      0      0      0  Master Error Table
N10:160        0      0      0      0      0      0      0      0      0      0
N10:170        0      0      0      0      0      0      0      0      0      0
N10:180        0      0      0      0      0      0      0      0      0      0
N10:190        0      0      0      0      0      0      0      0      0      0

N10:200      200    201    202    203      0      0      0      0      0      0  Write Data Block
N10:210      210    211    212    213      0      0      0      0      0      0  to Module
N10:220      220    221    222    223      0      0      0      0      0      0  Reg 200 to 249
N10:230        0      0      0      0      0      0      0      0      0      0
N10:240        0      0      0      0      0      0      0      0      0      0
```

SLC Ladder Logic

Master Mode Example #2 : Master Mode w/ Command Control Enabled

```
3150-MCM Master Example #2
Program Listing                    Processor File: MCM3EX2M.ACH                        Rung 3:0

Rung 3:0
| TRANSFER  |TRANSFER                TEST IF A              CALCULATE                              |
| ENABLE    |DONE                    DATA BLOCK             INDIRECT                               |
| (required)|(required)                                     OFFSET                                 |
|    I:1        O:1             +LES---------------+   +CPT---------------+                        |
|----] [--------]/[-----------+-+LESS THAN         +-+-+COMPUTE           +-+--------------------+-|
|       0          0          | |Source A    M1:1.0| | |Dest         N7:40| |                    | |
|                             | |                 *| | |                 1| |                    | |
|      Decode the BTR Block ID| |Source B     N7:20| | |Expression        | |                    | |
|      numbers which will be  | |                 4| | |M1:1.0 * 50       | |                    | |
|      returned from the module +------------------+ | +------------------+ |                    | |
|      and copy the data buffer                      |      COPY DATA       |                    | |
|      into the data table. This                     |      FROM            |                    | |
|      method of indirect addressing                 |      MODULE          |                    | |
|      is more efficient than the branched           | +COP---------------+ |                    | |
|      method used in Master Example #1              +-+COPY FILE         +-+                    | |
|                             |                        |Source     #M1:1.2|                      | |
|                             |                        |Dest  #N10:[N7:40]|                      | |
|                             |                        |Length           50|                     | |
|                             |                        +------------------+                      | |
|                             |                                                  ENCODES         | |
|                             |                                                  BLOCK           | |
|      Transfer the BTW Block ID                                            +MOV---------------+ | |
|      value from the read buffer ------------------------------------------+MOVE            +-+ |
|      (word 1) to the write buffer                                         |Source      M1:1.1| | |
|      (word 0) to setup the BTW cycle                                      |                 *| | |
|                             |                                             |Dest        M0:1.0| | |
|                             |                                             |                 *| | |
|                             |                                             +------------------+ | |
|                             | USER CFG                                         ENCODES         | |
|                             | DOWNLOAD                                         BLOCK           | |
|                             | SELECT                                                           | |
|       Test if the User wants|      B3                                     +MOV---------------+ | |
|       to re-configure the   +----] [--------------------------------------+MOVE            +-+ |
|       module, and if so then        0                                     |Source         255|   |
|       place a 255 into the BTW                                            |                  |   |
|       Block ID position.                                                  |Dest        M0:1.0|   |
|                                                                           |                 *|   |
|                                                                           +------------------+   |

Rung 3:1
COMMAND CONTROL MODE (DELETE IF NOT USING COMMAND CONTROL MODE)
If the BTR Block ID value is not that of a pass-thru command, then the CMD_EN
and CMD_DONE/ERR bits are copied and the subroutine to manipulate the bits is
called.
| TRANSFER  |TRANSFER        TEST IF A                                           COPY              |
| ENABLE    |DONE            DATA BLOCK                                          CMD_DN            |
| (required)|(required)                                                         CMD_ERR           |
|                                                                              STATUS            |
|    I:1        O:1     +LES---------------+                                +COP---------------+   |
|----] [--------]/[-----+LESS THAN         +------------------------------+-+COPY FILE         +-+-|
|       0          0    |Source A    M1:1.0|                              | |Source    #M1:1.52| | |
|                       |                 *|                              | |Dest        #B9:6| | |
|                       |Source B       256|                              | |Length          12| | |
|                       |                  |                              | +------------------+ | |
|                       +------------------+                              |                      | |
|                                                                         |      COMMAND         | |
|                                                                         |      CONTROL         | |
|                                                                         |      MODE            | |
|         The rung handles the logic for the                              |      HANDLER         | |
|         Command Control Mode.  The first branch                         | +JSR---------------+ | |
|         copies the Done and Error status bits, which                    +-+JUMP TO SUBROUTINE+-+ |
|         are being returned from the module, into a working              | |SBR file number  4| | |
|         file. The SBR 4 processes the Done and Error bits               | +------------------+ | |
|         for the commands which are in the Command Control Mode          |      COPY            | |
|         and clears the Enable bits if appropriate.                      |      CMD_EN          | |
|         The last branch transfers up the Enable bits.                   |      BITS            | |
|                                                                         | +COP---------------+ | |
|                                                                         +-+COPY FILE         +-+ |
|                                                                           |Source       #B9:0|   |
|                                                                           |Dest      #M0:1.51|   |
|                                                                           |Length           6|   |
|                                                                           +------------------+   |
```

SLC Ladder Logic

Master Mode Example #2 : Master Mode w/ Command Control Enabled

```
3150-MCM Master Example #2
Program Listing                          Processor File: MCM3EX2M.ACH                             Rung 3:2
Rung 3:2
WRITE DATA, COMMAND LIST OR CONFIGURATION BLOCK TO MODULE
Based on the value in the BTW Block ID, either the data or the command list is
moved to the module, or configuration parameters are moved to the module.  To
move additional data, add new branches
| TRANSFER        TRANSFER          DECODES                                     TRANSFER                     |
| ENABLE          DONE              BT WRITE                                    TO MODULE                    |
| (required)      (required)        BLOCK ID                                                                 |
|    I:1             O:1       +EQU---------------+                        +COP---------------+              |
|----] [--------+----]/[-----+-+EQUAL             +------------------------+COPY FILE         +-+----------+-|
|       0       |       0    | |Source A    M0:1.0|                        |Source    #N10:200| |          | |
|  Move the data for the module|                 *|                        |Dest       #M0:1.1| |          | |
|  addresses 200 to 249.  In | |Source B          4|                        |Length          50| |          | |
|  a Master application, these |                  |                        +------------------+ |          | |
|  register locations will be| +------------------+                                             |          | |
|  used to write values to   |      DECODES                                     TRANSFER        |          | |
|  the slaves.  |            |      BT WRITE                                    TO MODULE       |          | |
|               |            |      BLOCK ID                                                    |          | |
|  Command List |            | +EQU---------------+                        +COP---------------+ |          | |
|  Block IDs 80 and 81       +-+EQUAL             +------------------------+COPY FILE         +-+          | |
|  (through 99) are used to  | |Source A    M0:1.0|                        |Source      #N7:50| |          | |
|  transfer the Command List | |                 *|                        |Dest       #M0:1.1| |          | |
|  to the module. Each ‘block’ |Source B         80|                        |Length          50| |          | |
|  moves 5 commands to the   | |                  |                        +------------------+ |          | |
|  module.      |            | +------------------+                                             |          | |
|               |            |      DECODES                                     TRANSFER        |          | |
|               |            |      BT WRITE                                    TO MODULE       |          | |
|               |            |      BLOCK ID                                                    |          | |
|               |            | +EQU---------------+                        +COP---------------+ |          | |
|               |            +-+EQUAL             +------------------------+COPY FILE         +-+          | |
|               |            | |Source A    M0:1.0|                        |Source     #N7:100| |          | |
|               |            | |                 *|                        |Dest       #M0:1.1| |          | |
|               |            | |Source B         81|                        |Length          50| |          | |
|               |            | |                  |                        +------------------+ |          | |
|               |            | +------------------+                                             |          | |
|               |            |      DECODES                TRANSFER                             |          | |
|               |            |      BT WRITE               TO MODULE                            |          | |
|               |            |      BLOCK ID                                                    |          | |
|  Configuration of the      | +EQU---------------+   +COP---------------+                      |          | |
|  module       |            +-+EQUAL             +-+-+COPY FILE         +-+--------------------+          | |
|               |              |Source A    M0:1.0| | |Source       #N7:0| |                               | |
|               |              |                 *| | |Dest       #M0:1.1| |                               | |
|               |              |Source B       255| | |Length          40| |                               | |
|               |              |                  | | +------------------+ |                               | |
|               |              +------------------+ |                      |                               | |
|               |                                    | USER CFG             |                               | |
|               |                                    | DOWNLOAD             |                               | |
|               |                                    | SELECT               |                               | |
|               |                                    |      B3              |                               | |
|               |                                   +----(U)---------------+                               | |
|               |                                           0                                              | |
|               |                                                                               TRANSFER   | |
|               |                                                                               DONE       | |
|               |                                                                               (required) | |
|               |                                                                                          | |
|               |                                                                                  O:1     | |
|               +----------------------------------------------------------------------------------( )-----+ |
|                                                                                                     0      |

Rung 3:3
|                                                                                                            |
|----------------------------------------------------+END+---------------------------------------------------|
|                                                                                                            |
```

SLC Ladder Logic

Master Mode Example #2 : Master Mode w/ Command Control Enabled

```
3150-MCM Master Example #2
Program Listing                     Processor File: MCM3EX2M.ACH                            Rung 4:0

Rung 4:0
COMMAND CONTROL EXAMPLE LOGIC (DELETE IF NOT USING COMMAND CONTROL MODE)
The following rungs of logic control the unlatching of hte Command Enable bits
when the command is done successfully.
|                                                             LOGIC                   CMD EN BIT   |
|                                                             CMD ENABLE              CMD 1        |
|                                                             CMD 1                                |
|                                                                  B3                      B9      |
|-----------------------------------------------------------+----] [---------------------(L)-----+-|
|                                                           |      16                      0    | |
|                                                           | CMD DN BIT   CMD EN BIT            | |
|                                                           | CMD 1        CMD 1                 | |
|               This logic, typical for any Control         |      B9           B9               | |
|               Enabled command, takes the logical enable   +----] [-----+----(U)-----+----------+ |
|               which would be provided by application             96    |       0    |            |
|               ladder (B3/16) and latches the Enable.                   | LOGIC      |            |
|               When the Done bit is received, the enables               | CMD ENABLE |            |
|               are cleared. NOTE THAT THE ENABLE IS ONE SHOT            | CMD 1      |            |
|               IN THE MODULE.  THE MODULE MUST SEE A TRANSITION         |      B3    |            |
|               FROM 1 to 0 BEFORE IT WILL RE-ENABLE A COMMAND.          +----(U)-----+            |
|                                                                               16                 |

Rung 4:1
|                                                             LOGIC                   CMD EN BIT   |
|                                                             CMD ENABLE              CMD 2        |
|                                                             CMD 2                                |
|                                                                  B3                      B9      |
|-----------------------------------------------------------+----] [---------------------(L)-----+-|
|                                                           |      17                      1    | |
|                                                           | CMD DN BIT   CMD EN BIT            | |
|                                                           | CMD 2        CMD 2                 | |
|                                                           |      B9           B9               | |
|                                                           +----] [-----+----(U)-----+----------+ |
|                                                                  97    |       1    |            |
|                                                                        | LOGIC      |            |
|                                                                        | CMD ENABLE |            |
|                                                                        | CMD 2      |            |
|                                                                        |      B3    |            |
|                                                                        +----(U)-----+            |
|                                                                               17                 |

Rung 4:2
|                                                             LOGIC                   CMD EN BIT   |
|                                                             CMD ENABLE              CMD 3        |
|                                                             CMD 3                                |
|                                                                  B3                      B9      |
|-----------------------------------------------------------+----] [---------------------(L)-----+-|
|                                                           |      18                      2    | |
|                                                           | CMD DN BIT   CMD EN BIT            | |
|                                                           | CMD 3        CMD 3                 | |
|                                                           |      B9           B9               | |
|                                                           +----] [-----+----(U)-----+----------+ |
|                                                                  98    |       2    |            |
|                                                                        | LOGIC      |            |
|                                                                        | CMD ENABLE |            |
|                                                                        | CMD 3      |            |
|                                                                        |      B3    |            |
|                                                                        +----(U)-----+            |
|                                                                               18                 |
Rung 4:3
|                                                                                                  |
|-----------------------------------------------+END+----------------------------------------------|
```