3100/3150 – MDA-4
MDA Scientific CM4
Master Module
Revision 1.0

USER MANUAL

May 1997

ProSoft Technology, Inc.
9801 Camino Media
Suite 105
Bakersfield, CA 93311
prosoft@prosoft-technology.com
http://www.prosoft-technology.com

Please Read This Notice

Successful application of the MDA-4 module requires a reasonable working knowledge of the Allen-Bradley PLC/SLC hardware and the application in which the combination is to be used. For this reason, it is important that those responsible for implementing the MDA-4 satisfy themselves that the combination will meet the needs of the application without exposing personnel or equipment to unsafe or inappropriate working conditions.

This manual is provided to assist the user. Every attempt has been made to assure that the information provided is accurate and a true reflection of the product's installation requirements. In order to assure a complete understanding of the operation of the product, the user should read all applicable Allen-Bradley documentation on the operation of the A-B hardware.

Under no conditions will ProSoft Technology, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of the product.

Information in this manual is subject to change without notice and does not represent a commitment on the part of ProSoft Technology, Inc. Improvements and/or changes in this manual or the product may be made at any time. These changes will be made periodically to correct technical inaccuracies or typographical errors.

Quick Start Implementation Guide

Integration of the 3100/3150-MDA-4 module into a PLC/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 that will impact the long term success of the installation.

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

- Configure the slot as follows:

Other	ID Code xxxxx
Input File Length	8
Output File Length	8
Scanned Input File Length	8
Scanned Output File Length	8
M0 File Length	64
M1 File Length	64

a) Starting with one of the ladder logic programs provided on disk with the MDA-4 complete the following steps:

PLC 5	MDA4
SLC 5/03	MDA4_503 (See Appendix for SLC programming tips)

b) Edit the ladder logic provided on disk as needed for the application
   Verify rack and slot location in program
   Modify ladder instruction addresses as needed

c) Setup the Communication Configuration parameters
   Determine each port's communication configuration requirements

d) Setup the Polling List for each port

e) Identify the jumper requirements (See Appendix)

f) Make up the communication cables

g) Place processor into the run mode

h) Monitor the data table for the Data and Error Status values

Product Revision History

06/01/97	Revision 1.0 Initial release of product

Table of Contents

1 Product Specifications

The 3100/3150-MDA-4 ("MDA Scientific CM4 Master Module") product family allows Allen-Bradley 1771 and 1746 I/O compatible processors to easily interface as a host with MDA Scientific CM4 gas monitoring hardware (See 3100/3150-MDA-16 for a System 16 solution).

1.1 Operating Specifications

The MDA-4 product includes the following standard features:

- Two fully configurable serial ports, each capable of supporting the CM4 Master functionality
- Supports up to x CM4 units per serial port
- Support movement of binary, integer, ASCII, and floating point data types
- Memory mapping will be pre-defined in the module to ease implementation in the ladder program
- RS-485 connection from each port directly to the CM4 units
- Software configuration (From processor ladder logic)

Slave Addr	:	0 to 31
Command	:	Select command to be executed
Char Size	:	8 bits (fixed)
Parity	:	None (fixed)
Stop Bit	:	1 (fixed)
Baud Rate	:	300 TO 9,600
RTS to TxD	:	50 ms (fixed)
Timeout	:	1 second
Polling Rate	:	1 second (fixed)

- Response time
  The protocol drivers are written in Assembly and in a compiled higher level language. As such, the interrupt capabilities of the hardware are fully utilized to minimize delays, and to optimize the product's performance
- Supported CM4 command codes:

Read Comands

0x30	Get System Information
0x31	Get Unit Status
0x36	Get Alarm History
0x37	Get Current Point Status

Write Commands

0x51	Reset Fault or Alarm
0x52	Set Key-Code
0x53	Lock Keyboard
0x60	End Point Lock-On
0x61	Start Point Lock-On

- Operating Mode returned to ladder processor
- Error Codes returned to the ladder processor

1.2 Hardware Specifications

- Backplane Current Load :

3100	: 0.65 A
3150	: 0.15 A at 5 V
	0.04 A at 24 V

- Operating Temperature : 0 to 60 °C
- Storage Temperature : -40 to 85 °C
- Connections :

3100	: 2 - DB25 Female Connectors
3150	: 2 - DB9 Male Connectors

2 Writing Data to the Module

Data transfers between the processor and the ProSoft Technology module occur using the Block Transfer commands, in the case of the PLC, and M0/M1 data transfer commands, in the case of the SLC. These commands transfer up to 64 physical registers per transfer. The logical data length changes depending on the data transfer function.

The following discussion details the data structures used to transfer the different types of data between the ProSoft Technology module and the processor. The term 'Block Transfer' is used generically in the following discussion to depict the transfer of data blocks between the processor and the ProSoft Technology module. Although a true Block Transfer function does not exist in the SLC, we have implemented a pseudo-block transfer command in order to assure data integrity at the block level. Examples of the PLC and SLC ladder logic are included in Appendix A.

> In order for the ProSoft Technology module to function, the PLC must be in the RUN mode, or in the REM RUN mode. If in any other mode (Fault/PGM), the block transfers between the PLC and the module will stop, and communications will halt until block transfers resume.

2.1 Block Transferring Data to the Module

Data transfer to the module from the processor is executed through the Block Transfer Write function. The different types of data which are transferred require slightly different data block structures, but the basic data structure is:

Word	Name	Description
0	BTW Block ID	A block page identifier code. This code is used by the ProSoft module to determine what to do with the data block. Valid codes are: BTW Code — Description 0-1 — Command Control and Data 255 — Module Communication Configuration
1 to 63	Data	The data to be written to the module. The structure of the data is dependent on the Block ID code. The following sections provide details on the different structures.

> Although the full physical 64 words of the data buffer may not be used, the BTW and M0 lengths must be configured for 64 words, otherwise module operation will be unpredictable.

2.1.1 Communications Configuration [BTW Block ID 255]

The ProSoft Technology firmware communication parameters must be configured at least once when the card is first powered up, and any time thereafter when the parameters must be changed.

Power Up

On power up, the module enters into a logical loop waiting to receive configuration data from the processor. While waiting, the module sets the second word of the BTR buffer (the BTW Block ID) to 255, telling the processor that the module must be configured before anything else will be done. The module will continuously perform block transfers until the communications configuration parameters block is received. Upon receipt, the module will begin execution of the command list if present, or begin looking for the command list from the processor.

Changing parameters during operation

Changing values in the configuration table can be done at any time. The module does not accept any of the changes until the 're-configuration' process is initiated. This can be accomplished in several ways, including:

1. Cycle power to the rack
2. Press the reset pushbutton on the module (3100 only)
3. Move 255 into BTW Block ID position (See example logic when B3/0 is set)

During this process, the 'CFG' LED will toggle, giving a visual indication that the module has received the configuration block.

Transferring the Communications Configuration Parameters to the module will force a reset of the communication port, as well as dropping DTR for 200 ms pulses to reset any attached hardware.

The configuration data block structure which must be transferred from the processor to the module is as follows:

BTW Buffer	Example Data Addr	Name
0		BTW Block ID
		Port / Module Configuration
1	N7:0	Baud Rate – Port 1
2	N7:1	Baud Rate – Port 2
3	N7:2	Number of Active Slaves
4	N7:3	Spare
5	N7:4	Spare
6	N7:5	Spare
7	N7:6	Spare
8	N7:7	Spare
9	N7:8	Spare
10	N7:9	Spare
		Pollling List / Port Select
11	N7:10	Slave 1
12	N7:11	Slave 2
13	N7:12	Slave 3
14	N7:13	Slave 4
15	N7:14	Slave 5
16	N7:15	Slave 6
17	N7:16	Slave 7
18	N7:17	Slave 8
19	N7:18	Slave 9
20	N7:19	Slave 10

The structure of the Port and Module Configuration Data block, and the meaning of each of the configuration parameters is outlined in the following table.

Data Addr	Name	Description
N7:0 N7:1	Baud Rate – Port 1 Baud Rate - Port 2	The baud rate at which the port is to operate. The available configurations are as follows: Value — Baud Rate 0 — 300 Baud 1 — 600 Baud 2 — 1200 Baud 3 — 2400 Baud 4 — 4800 Baud 5 — 9600 Baud
N7:2	Number of Active Slaves	This value should represent the total number of slaves which this module will be polling between the two ports. The module will support up to 10 CM4 units between the two ports. If not all 10 slots are being used, the operation of the module can be optimized by accurately selecting the number of slaves. The optimization comes primarity from reduced number of data block transfers. Valid values range from 0 to 10. If 0 is configured the module assumes that all 10 slots are active.

The Polling List / Port Select data structure for the module is outlined in the following table. These configuration values are used to select the slave address for each of the 10 possible slaves (i.e., what device addresses to poll for data), and to select which port to poll the slave address on.

Data Addr	Name	Description
N7:10 N7:11 N7:12 N7:13 N7:14 N7:15 N7:16 N7:17 N7:18 N7:19	Slave Address – Low Byte Port Select – High Byte	This is a high byte / low byte type of selection. The Port Select byte (high byte) is configured as follows: **Value** — **Description** 0 — Port 1 selected 1 — Port 2 selected The Slave Address byte (low byte) is configured as follows: **Value** — **Description** 0 — Inactive – (Master Address) 1-255 — Slave Device Address Example : N7:10 3 Poll Slave #3 on Port 1 N7:11 257 Poll Slave #1 on Port 2 = 256(Port 2)+ Slave Address

2.1.2 Writing Control Block to Module[BTW Block ID 0 and 1]

The BTW Block ID 0 and 1 blocks are used to transfer Command Control/Data information to the module for the 10 possible slaves. This data is used by the module to determine which Read and Write protocol commands to execute, as well as what data values to write to the slave.

The following tables show how the data for all 10 slaves is moved up into the module. The example ladder logic in the Appendix also shows by example the ladder logic needed to implement this functionality.

BTW Buffer	Example Data Addr	Name
0		BTW Block ID = 0
1 to 10	N9:0 To N9:9	Slave #1 Command Block
11 to 20	N9:10 To N9:19	Slave #2 Command Block
21 to 30	N9:20 To N9:29	Slave #3 Command Block
31 to 40	N9:30 To N9:39	Slave #4 Command Block
41 to 50	N9:40 To N9:49	Slave #5 Command Block

BTW Buffer	Example Data Addr	Name
0		BTW Block ID = 1
1 to 10	N9:50 To N9:59	Slave #6 Command Block
11 to 20	N9:60 To N9:69	Slave #7 Command Block
21 to 30	N9:70 To N9:79	Slave #8 Command Block
31 to 40	N9:80 To N9:89	Slave #9 Command Block
41 to 50	N9:90 To N9:99	Slave #10 Command Block

Data Addr	Name	Description
N9:0 N9:10 N9:20 N9:30 N9:40 N9:50 N9:60 N9:70 N9:80 N9:90	Read Command Enable Bits Slaves 1 to 10	This register is a bit mapped set of enable bits that will allow the application programmer to control the execution of the read commands, and therefore the relative update timing. **<u>Bit</u>** **<u>Description</u>** 0 Get System Information – 0x30 1 Get Unit Status – 0x31 2 Get Point #1 Status – 0x37 3 Get Point #2 Status – 0x37 4 Get Point #3 Status – 0x37 5 Get Point #4 Status – 0x37 6 Get Alarm History – 0x36 7 Get Point Cfg #1 – 0x35 8 Get Point Cfg #2 – 0x35 9 Get Point Cfg #3 – 0x35 10 Get Point Cfg #4 – 0x35
N9:1 N9:11 N9:21 N9:31 N9:41 N9:51 N9:61 N9:71 N9:81 N9:91	Write Command Enable Bits Slaves 1 to 10	This register is a bit mapped set of enable bits that will allow the application programmer to control the execution of the write commands to a slave. **<u>Bit</u>** **<u>Description</u>** 0 Alarm/Fault Reset – 0x51 1 Lock Keyboard – 0x53 2 Start Point #1 Lock – 0x61 3 Start Point #2 Lock – 0x61 4 Start Point #3 Lock – 0x61 5 Start Point #4 Lock – 0x61 6 End Point Lock – 0x60
N9:2 N9:12 N9:22 N9:32 N9:42 N9:52 N9:62 N9:72 N9:82 N9:92	Alarm/Fault Reset Selection Slaves 1 to 10	This register is a bit mapped set of bits that will allow the application programmer to control which point alarms are cleared and if the faults are to be cleared. **<u>Bit</u>** **<u>Description</u>** 0 Point 1 Alarms 1 Point 2 Alarms 2 Point 3 Alarms 3 Point 4 Alarms 4 Faults
N9:3 N9:13 N9:23 N9:33 N9:43 N9:53 N9:63 N9:73 N9:83 N9:93	Lock Keyboard Command Slaves 1 to 10	This register is a bit mapped set of enable bits that will allow the application programmer to control the locking/unlocking of the keyboard. This selection is used in conjunction with the configurable keycode parameter in the next register. **<u>Bit</u>** **<u>Description</u>** 0 0 = Unlocked 1 = Locked
N9:4 N9:14 N9:24 N9:34 N9:44 N9:54 N9:64 N9:74 N9:84 N9:94	Lock Keyboard Keycode Slaves 1 to 10	This register is a bit mapped set of enable bits that will allow the application programmer to control the execution of the write commands to a slave. **<u>Value</u>** **<u>Description</u>** 0 to 9999 Keycode entry sent to CM4

The following diagram shows the relative positioning of the data structure in the example ladder logic. Note that any data file can be used in an application. Simply changing the mapping of the COP commands in the example ladder logic will account for any file selection.

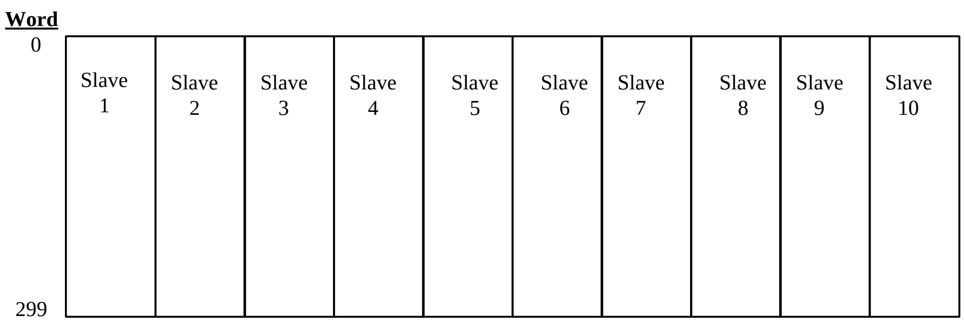
	Read Cmd Enable	Write Cmd Enable	Alarm Fault Reset Select	Lock Cmd	Keyboard Keycode						
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	
N9:0	0	0	0	0	0	0	0	0	0	0	Slave #1 Command Control
N9:10	0	0	0	0	0	0	0	0	0	0	Slave #2 Command Control
N9:20	0	0	0	0	0	0	0	0	0	0	Slave #3 Command Control
N9:30	0	0	0	0	0	0	0	0	0	0	Slave #4 Command Control
N9:40	0	0	0	0	0	0	0	0	0	0	Slave #5 Command Control
N9:50	0	0	0	0	0	0	0	0	0	0	Slave #6 Command Control
N9:60	0	0	0	0	0	0	0	0	0	0	Slave #7 Command Control
N9:70	0	0	0	0	0	0	0	0	0	0	Slave #8 Command Control
N9:80	0	0	0	0	0	0	0	0	0	0	Slave #9 Command Control
N9:90	0	0	0	0	0	0	0	0	0	0	Slave #10 Command Control

3 Reading From the Module

This section provides reference level details on the transfer of data from the PLC/SLC processor to the module.

3.1 Transferring data from the module

When the Master port driver reads data from a slave the resulting data is placed into the ProSoft module's data space. This data space is broken down into ten(10) 300 word data blocks, with each 300 word block representing the data from one(1) slave. The following diagram shows this structure:

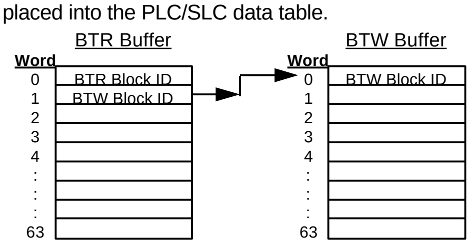


In order to get this data into the PLC/SLC, the blocks are broken down into 50 word 'pages' and transferred to the ladder logic across the backplane using the standard BTR or M1 instructions. The following sections detail the structure of this data and the mechanism by which all of the data is transferred.

Although the full physical 64 words of the data buffer may not be used, the BTR and M1 lengths must be configured for a length of 64 words, otherwise module operation will be unpredictable

3.1.1 The Read Data Block Structure

The BTR buffer definition is:

Word	Name	Description
0	BTR Block ID	The ladder logic uses this value to determine the contents of the data portion of the BTR buffer. With some conditional testing in ladder logic, the data from the module can be placed into the PLC/SLC data table.  The relationship between the BTR Block ID number and the register table can be put into an equation: Starting Register Address = Block ID Number * 50 Valid codes are between 0 and 59 (Each slave will consume up to 6 blocks).

(Continued)

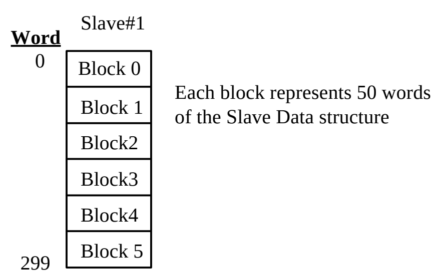
Word	Name	Description
1	BTW Block ID	The module returns this value to the processor to be used to enable the movement of Command data to the module. The BTW Block ID number is developed by the module. Valid codes are: **BTW Code** — **Description** 0-1 — Command Data 255 — Module Configuration
2 to 51	Data	This data will contain data received from the slaves. The values will be 16 bit register values, and should be placed into integer files. Note that the user application ladder logic controls the placement and use of the data registers.

3.2 Reading Data from the Module[BTR Block ID 0 and 59]

In order to understand the movement of data from the module to the ladder memory, it is important to understand the building of the memory map in the module. Shown earlier in the diagram above is that fact that the module stores the Slave Data in individual 300 word blocks.

The transfer of this data is accomplished by breaking each of the 300 word blocks down into six(6) 50 words blocks. These individual 50 word blocks are 'paged' across the backplane within the BTR Buffer structure discussed above. Using the BTR Block ID number, the ladder logic is able to determine where to place the data in the ladder logic memory.

The following diagram shows the Slave #1 Data block broken down into its 50 word blocks, and the corresponding BTR Block ID number for each of the blocks.



The following table shows the BTR Block ID numbering for all 10 slaves:

Words	Slave 1	Slave 2	Slave 3	Slave 4	Slave 5	Slave 6	Slave 7	Slave 8	Slave 9	Slave 10
0 to 49	0	6	12	18	24	30	36	42	48	54
50 to 99	1	7	13	19	25	31	37	43	49	55
100 to 149	2	8	14	20	26	32	38	44	50	56
150 to 199	3	9	15	21	27	33	39	45	51	57
200 to 249	4	10	16	22	28	34	40	46	52	58
250 to 299	5	11	17	23	29	35	41	47	53	59

3.2.1 The Slave Data Block Structure

The data structure for each slave is predefined and was developed during the development of the module. As discussed above, the individual slave data is stored in a 300 word data block. The structure of the data block is as follows:

Data Addr Offset	Name	Description
0	Communication Counter	This value represents a 0 to 32767 rollover counter that increments each time communication with the slave occurs. Incrementing is independent of the command executed.
1	Communicaton Status Error	This register is used to indicate that status of communications between the module and the particular slave. A non-zero number indicates the type of communicatoin problem which is occuring. This value is not latched and will therefore clear to 0 on the first successful communications. The values which can be expected in the field are: **Value** — **Description** 0 — All OK 1 — TBD 2 — TBD 3 — Error in Response 8 — Timeout Error 16 — Module Config Error 254 — Checksum Error 255 — TX Fail (Verify RTS/CTS jumper)
2	Read Command Done Bits	These bits indicate the execution of the particular command. The module will clear the bits immediately after the block transfer to assure that they are not held on. **Value** — **Description** 0 — Get System Information – 0X30 1 — Get Unit Status – 0x31 2 — Get Point #1 Status – 0x37 3 — Get Point #2 Status – 0x37 4 — Get Point #3 Status – 0x37 5 — Get Point #4 Status – 0x37 6 — Get Alarm History – 0x36 7 — Get Point Cfg #1 – 0x35 8 — Get Point Cfg #2 – 0x35 9 — Get Point Cfg #3 – 0x35 10 — Get Point Cfg #4 – 0x35
3	Write Command Done Bits	These bits indicate the execution of the particular write command. The module will clear its bit image immediately after the block transfer to assure that they are cleared during the subsequent block transfers. **Value** — **Description** 0 — Alarm/Fault Reset – 0x51 1 — Lock Keyboard – 0x53 2 — Start Point #1 Lock – 0x61 3 — Start Point #2 Lock – 0x61 4 — Start Point #3 Lock – 0x61 5 — Start Point #4 Lock – 0x61 6 — End Point Lock – 0x60
4	Alarm Reset Status	Value returned from a 0x51
5	Lock Keyboard Status	Value returned from a 0x53
6	End Point Lock-on Results	Value returned from a 0x60
7	Start Point Lock-on Results	Value returned from a 0x61
8	Spare	
9	Spare	

Data Addr Offset	Name	Description
10 11 12	Month Day Year	Date returned from the instrument during the last execute read command. Each read command returns the date, so these registers should continuously be getting updated by the CM4.
13 14 15	Hour Minute Second	Time returned from the instrument during the last execute read command. Each read command returns the time, so these registers should continuously be getting updated by the CM4.
16	Number of Alarms	Value returned from a 0x36 – Get Alarm History command.
17	Spare	
18	Spare	
19	Spare	
20	Serial Num	
21	software rev	
22	VIP	
23	Prom csum msb	
24	Prom csum lsb	
25	Status - read verified	
26	General Status	
27	Flash Mem remaining	
28	Chemcassette windows remaining	
29	Chemcassette days remaining	
30	Internal Filter Days in Use	
31	External Filter Days in Use	
32	Flow Rate Point 1	
33	Flow Rate Point 2	
34	Flow Rate Point 3	
35	Flow Rate Point 4	
36	Status - Optics Cal and Maint Status	Optics in High byte and Maintenance in Low byte
37	Spare	
38	Spare	
39	Spare	
40	MDA Gas Abbrev	Pt 1
41	MDA Gas Abbrev	Pt 1
42	MDA Gas Abbrev	Pt 1
43	Format code	Pt 1
44	Flow Rate - Current Flow	Pt 1
45	TWA Start Data	Pt 1
46	TWA Start Time	Pt 1
47	TWA End Date	Pt 1
48	TWA End Time	Pt 1
49	TWA Concentration	Pt 1
50	Last Concentration	Pt 1
51	Alarm Status	Pt 1
52	Point Status	Pt 1
53	Point Cfg Status	Pt 1
54	Alarm Level 1	Pt 1
55	Alarm Level 2	Pt 1
56	Spare	Pt 1
57	Spare	Pt 1
58	Spare	Pt 1
59	Spare	Pt 1
60	MDA Gas Abbrev	Pt 2
61	MDA Gas Abbrev	Pt 2
62	MDA Gas Abbrev	Pt 2
63	Format code	Pt 2
64	Flow Rate - Current Flow	Pt 2
65	TWA Start Data	Pt 2
66	TWA Start Time	Pt 2

67	TWA End Date	Pt 2
68	TWA End Time	Pt 2
69	TWA Concentration	Pt 2
70	Last Concentration	Pt 2
71	Alarm Status	Pt 2
72	Point Status	Pt 2
73	Point Cfg Status	Pt 2
74	Alarm Level 1	Pt 2
75	Alarm Level 2	Pt 2
76	Spare	Pt 2
77	Spare	Pt 2
78	Spare	Pt 2
79	Spare	Pt 2
80	MDA Gas Abbrev	Pt 3
81	MDA Gas Abbrev	Pt 3
82	MDA Gas Abbrev	Pt 3
83	Format code	Pt 3
84	Flow Rate - Current Flow	Pt 3
85	TWA Start Data	Pt 3
86	TWA Start Time	Pt 3
87	TWA End Date	Pt 3
88	TWA End Time	Pt 3
89	TWA Concentration	Pt 3
90	Last Concentration	Pt 3
91	Alarm Status	Pt 3
92	Point Status	Pt 3
93	Point Cfg Status	Pt 3
94	Alarm Level 1	Pt 3
95	Alarm Level 2	Pt 3
96	Spare	Pt 3
97	Spare	Pt 3
98	Spare	Pt 3
99	Spare	Pt 3
100	MDA Gas Abbrev	Pt 4
101	MDA Gas Abbrev	Pt 4
102	MDA Gas Abbrev	Pt 4
103	Format code	Pt 4
104	Flow Rate - Current Flow	Pt 4
105	TWA Start Data	Pt 4
106	TWA Start Time	Pt 4
107	TWA End Date	Pt 4
108	TWA End Time	Pt 4
109	TWA Concentration	Pt 4
110	Last Concentration	Pt 4
111	Alarm Status	Pt 4
112	Point Status	Pt 4
113	Point Cfg Status	Pt 4
114	Alarm Level 1	Pt 4
115	Alarm Level 2	Pt 4
116	Spare	Pt 4
117	Spare	Pt 4
118	Spare	Pt 4
119	Spare	Pt 4
120	Alarm 1	See Alarm Data block structure below
130	Alarm 2	See Alarm Data block structure below
140	Alarm 3	See Alarm Data block structure below
150	Alarm 4	See Alarm Data block structure below
160	Alarm 5	See Alarm Data block structure below
170	Alarm 6	See Alarm Data block structure below
180	Alarm 7	See Alarm Data block structure below
190	Alarm 8	See Alarm Data block structure below
200	Alarm 9	See Alarm Data block structure below

210	Alarm 10	See Alarm Data block structure below
220	Alarm 11	See Alarm Data block structure below
230	Alarm 12	See Alarm Data block structure below
240	Alarm 13	See Alarm Data block structure below
250	Alarm 14	See Alarm Data block structure below
260	Alarm 15	See Alarm Data block structure below
270	Alarm 16	See Alarm Data block structure below
280 to 289	Spare	
290 to 299	Product Information Structure	This data is only returned for Slave #1. See below for structure.

3.2.2 Alarm Data Block Structure

Up to 16 Alarm Data Blocks are returned from each CM4. The Alarm Data has been turned into a 10 word structure to allow viewing in the PLC/SLC data table to be easier. The structure of this data is as follows:

Note that the Alarm Data structure is shown only for Alarm #1. This structure repeats itself 16 times on 10 words offsets.

Data Addr Offset	Name	Description
120	Date Stamp	MDA format Date field (2 bytes packed with MDY)
121	Time Stamp	MDA format Time field (2 bytes packed with HMS)
122 123 124	Gas Abbreviation	
125	Point Number	Point number in alarm. **Value** **Description** 0 Point #1 1 Point #2 2 Point #3 3 Point #4
126	Format Code	
127	Concentration	
128	Alarm Level	Alarm level: **Value** **Description** 0 Level 1 1 Level 2
129	Spare	

3.2.3 Product Information Data Structure

Product revision information which may be useful during debuging and troubleshooting in the future is included in this data structure. This data block is only returned with the data from slave #1. Therefore it will be returned at the tail end of BTR Block ID 5. If all 300 words are being read from the unit, this data will automatically be included.

Data Addr Offset	Name	Description
290 291	Product Name	These two words represent the product name of the module in an ASCII representation. In the case of the 3750 product, the letters ‘ MDA4‘ should be displayed when placing the programming software in the ASCII data representation mode.
292 293	Product Revision	These two words represent the product revision level of the firmware in an ASCII representation. An example of the data displayed would be ‘1.00’ when placing the programming software in the ASCII data representation mode.
294	Product Operating System	This word represents the module’s internal operating system revision level in an ASCII representation.
295	Product Run Number	This number represents the ‘batch’ number that your particular chip belongs to in an ASCII representation.

4 Protocol Commands

The ProSoft Technology MDA-4 module Master driver supports several commands from the MDA-4 Command set.

4.1 MDA-4 Read Data Commands

The MDA-4 module supports a command subset of the Protocol Specification consisting primarily of the commands required to intialize and read data from several units. The following sections detail the different commands supported by the module.

4.1.1 0x30 – Get System Information

This command requests information about the slave system only and the software revision currently in use.

4.1.2 0x31 – Get Unit Status

This command requests the current conditoin or status of the salve. This command allows the master to inquire about the general operating condition of the system.

4.1.3 0x36 – Get Alarm History

This command queries the unit for any alarms. The unit saves only the 16 most recent alarms regardless of point. The alarms can all be on one point or there can be alarsm from several points.

4.1.4 0x37 – Get Current Point Status

The command queries an individual point for its current status.

4.1.5 0x35 – Get Point Configuration

The command queries an individual point for its current configuration.

4.2 MDA-4 Write and Control Commands

The MDA-4 module supports a command subset of the Protocol Specification consisting primarily of the commands required to intialize and read data from several units. The following sections detail the different

4.2.1 0x51 – Reset Faults or Alarms

This command allows a remote reset of any faults or alarm conditions.

4.2.2 0x53 – Lock Keyboard

This command allows the ladder program to lock out the keyboard. The keyboard can be disabled, preventing unauthorized user intervention by enabling the keypad lock out and sending a valid key code.

4.2.3 0x60 – End Point Lock-on

This command unlocks the unit from a single pont lock-on to all other points that are enabled. When this command is issued, a new TWA start for all points.

4.2.3 0x61 – Start Point Lock-on

This command locks the unit to one specific pont. When this command is issued, all other points are disabled and locked-on point continues to moniotr for concentration and TWA.

5 Diagnostics & Troubleshooting

Several hardware diagnostics capabilities have been implemented using the LED indicator lights on the front of the module. The following sections explain the meaning of the individual LEDs for both the PLC and the SLC platforms.

5.1 3100 PLC Platform

The following table documents the LEDs for the 3100-MDA-4 module.

ProSoft CIM Card

ACTIVE	O	O	FLT
CFG	O	O	BPLN
ERR1	O	O	ERR2
TXD1	O	O	TXD2
RXD1	O	O	RXD2

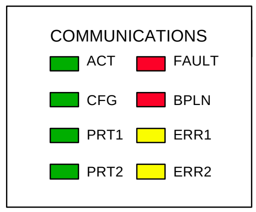
ProSoft CIM	A-B DB/B	Color	Status	Indication
ACT	ACT	Green	Blink (Fast)	Normal state : The module is operating normally and successfully Block Transferring with the PLC
			On	The module is receiving power from the backplane, but there may be some other problem
			Blink (1/Sec)	Indicates the module has somehow entered the Basic Programming Mode. Verify jumper JW4 (DB/B only) configuration. If all are correct, then contact the factory
			Off	The module is attempting to Block Transfer with the PLC and has failed. The PLC may be in the PGM mode or may be faulted
FLT	FLT	Red	Off	Normal State : No system problems are detected during background diagnostics
			On	A system problem was detected during background diagnostics. Please contact factory for technical support
CFG	DH485	Green	Off	Normal state : No configuration related activity is occurring at this time
			Blink	This light blinks every time a Module Configuration block (ID = 255) is received from the processor ladder logic
			On	The light is on continuously whenever a configuration error is detected. The error could be in the Port Configuration data or in the System Configuration data. See Section 4 for details
BPLN	BTLO	Red	Off	Normal State : When this light is off and the ACT light is blinking quickly, the module is actively Block Transferring data with the PLC
			On	Indicates that Block Transfers between the PLC and the module have failed.(Not activated in the initial release of the product)
ERR1 ERR2	LED1 LED2	Amber	Off	Normal State : When the error LED is off and the related port is actively transferring data, there are no communication errors
			Blink	Periodic communication errors are occurring during data communications.
			On	This LED will stay on under several conditions: • CTS input is not being satisfied • Port Configuration Error • System Configuration Error • Unsuccessful comm on MDA-4 slave • Recurring error condition on MDA-4 master

Tx1 Tx2	PT1X PT2X	Green	Blink	The port is transmitting data.
Rx1 Rx2	PT1R PT2R	Green	Blink	The port is receiving data

5.2 3150 SLC Platform

The following table documents the LEDs for the 3150-MDA-4 module.

3150-MDA-4



LED Name	Color	Status	Indication
ACT	Green	Blink (Fast)	<u>Normal state</u> : The module is operating normally and successfully Block Transferring with the SLC
		On	The module is receiving power from the backplane, but there may be some other problem
		Blink (1/Sec)	Indicates the module has somehow entered the Basic Programming Mode. Verify jumper JW3 (BAS only) configuration. If all are correct, then contact the factory
		Off	The module is attempting to Block Transfer with the SLC and has failed. The SLC may be in the PGM mode or may be faulted (*Not in initial release)*
FLT	Red	Off	<u>Normal State</u> : No system problems are detected during background diagnostics
		On	A system problem was detected during background diagnostics. Please contact factory for technical support
CFG	Green	Off	<u>Normal state</u> : No configuration related activity is occurring at this time
		Blink	This light blinks every time a Module Configuration block (ID = 255) is received from the processor ladder logic
		On	The light is on continuously whenever a configuration error is detected. The error could be in the Port Configuration data or in the System Configuration data. See Section 4 for details
BPLN	Red	Off	<u>Normal State</u> : When this light is off and the ACT light is blinking quickly, the module is actively Block Transferring data with the SLC
		On	Indicates that Block Transfers between the SLC and the module have failed
ERR1 ERR2	Amber	Off	<u>Normal State</u> : When the error LED is off and the related port is actively transferring data, there are no communication errors
		Blink	Periodic communication errors are occurring during data communications. See Section 4 to determine the error condition
		On	This LED will stay on under several conditions: • CTS input is not being satisfied • Port Configuration Error • System Configuration Error • Unsuccessful comm on MDA-4 slave • Recurring error condition on MDA-4 master
TxRx1 TxRx2	Green	Blink	The port is communicating, either transmitting or receiving data

5.3 Troubleshooting

In order to assist in the troubleshooting of the module, the following tables have been put together to assist you. Please use the following to help in using the module, but if you have additional questions or problems please do not hesitate to contact us.

The entries in this section have been placed in the order in which the problems would most likely occur after powering up the module.

Problem Description	Steps to take
BPLN light is on (SLC)	The BPLN light comes on when the module does not think that the SLC is in the run mode (i.e., SLC is in PGM or is Faulted). If the SLC is running then verify the following: • Verify the SLC Status File to be sure the slot is enabled • The Transfer Enable/Done Bits (I/O Bits 0 for the slot with the module) must be controlled by the ladder logic. See Section 2.x for details or the example ladder logic in the Appendix. • If the ladder logic for the module is in a subroutine file verify that there is a JSR command calling the SBR
CFG light does not clear after power up (no ERR LED)	The 255 BTW Block ID number is not being detected by the module. This could be due to a Block Transfer failure (PLC) or to an error in the ladder logic preventing the 255 value from being moved to the BTW buffer
CFG light does not clear after power up (w/ ERR LED)	If the BPLN light has been cleared, then several of the Port and System configuration values are value checked by the module to be sure that legal entries have been entered in the data table. Verify the Error Status Table for an indication of a configuration error.
CFG light toggles	Under normal conditions, the CFG LED will clear immediately after receipt. If the CFG light toggles, this usually indicates that the logic condition which places the 255 Block ID value in the BTW buffer is not being cleared. Check the ladder logic to be sure that the condition moving the 255 value is not held true.
Module is not transmitting	Presuming that the processor is in run, verify the following: • CTS input is not satisfied (check RTS/CTS jumper) • Check Error Status codes for 255 code. If so see next problem • If in slave mode, verify the slave address being requested from the Host • If in master mode, verify the command list configuration and that the Command List is being moved into the module (i.e., check the Command Block Cnt and associated ladder logic)
Error Code 255 in Status Table	This is caused by only one thing, a missing CTS input on the port. If a cable is connected to the port, then verify that a jumper has been installed between the RTS and CTS pins. If so then there may be a hardware problem.
Overwriting data blocks	This condition normally occurs when it is forgotten that the BTW Block ID value is being manipulated by the module, and that it always starts at 0. Please verify that the configuration of the module (Read and Write Block Counts) is not causing data from the PLC/SLC to overwrite data being returned from the module. A simple method for verifying this is to perform a histogram on the BTW Block ID register.
Data swapping is occurring (3100 only)	Under several circumstances data swapping in the module has occurred. This swapping has always been associated with the 8/16 pt jumper on the back of the card. Please verify that the jumper is in the 8pt position

Problem Description	Steps to take
New configuration values are not being accepted by the module	In order for new values to be moved to the module a Block Transfer Write with a Block ID of 255 must be transmitted to the module. The 'User Config Bit' in the example logic accomplishes this. In the example logic the bit must either be set in the data table manually or the module must be powered down/reset. In order to download the configuration upon transitioning from PGM to RUN, simply add a run to set the 'User Config Bit' based on the First Scan Status Bit (S1:1/15)
Error Codes being returned in locations with no commands (Master Configuration)	Be sure that the Command Block Count configuration value is setup correctly. There should be one branch of logic in the Write Rung corresponding to each Command Block to be written (i.e., a Command Block Count of 2 should have two branches of logic to handle BTW Block IDs 80 and 81. If the Command Block Count configuration value exceeds the number of branches in logic, the Command List is inadvertently being duplicated. To resolve the issue, either add more branches of logic or reduce the Command Block Count value to match the number of BTW logic branches.
RX1 or RX2 on continuously (3100 only)	The TX and RX LEDs on the module are tied to the hardware state of the ports (i.e., are not controlled directly by firmware). When the RX LED is on continuously is normally indicates that the polarity of the cable connection to the port is swapped. This is particularly true in RS-485 and RS-422 modes.

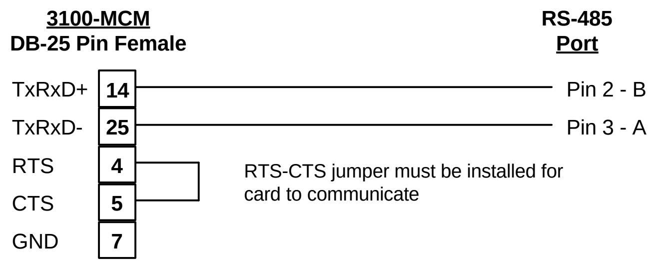
6 Cable Connections

The following diagrams show the connection requirements for the ports on the 3100 and 3150 modules.

3100 Module

RS-485/2-Wire Connection

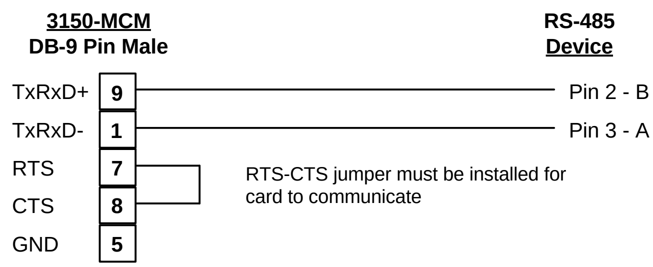
The jumper on the module must be set in the RS-485 position for all 2-wire applications



3150 Module

RS-485/2-Wire Connection

The jumper on the module must be set in the RS-485 position for all 2-wire applications



RS-485 and RS-422 Tip

If communication in the RS-422/RS-485 mode does not work at first, despite all attempts, try switching termination polarities. Some manufacturers interpret +/- and A/B polarities differently.

Appendix

A Support, Service and Warranty

Technical Support

ProSoft Technology survives on its ability to provide meaningful support to its customers. Should any questions or problems arise, please feel free to contact us at:

Factory/Technical Support
ProSoft Technology, Inc.
9801 Camino Media, Suite 105
Bakersfield, CA 93311
(661) 664-7208
(800) 326-7066
(661) 664-7233 (fax)
E-mail address: prosoft@prosoft-technology.com
Web Site : http://www.prosoft-technology.com

Before calling for support, please prepare yourself for the call. In order to provide the best and quickest support possible, we will most likely ask for the following information (you may wish to fax it to us prior to calling):

1. Product Version Number
2. Configuration Information
   - Communication Configuration
   - Master Command List
   - Jumper positions
3. System hierarchy
4. Physical connection information
   - RS-232, 422 or 485
   - Cable configuration
5. Module Operation
   - Block Transfers operation
   - LED patterns

An after-hours answering system (on the Bakersfield number) allows pager access to one of our technical and/or application support engineers at all times to answer any questions you have.

Module Service and Repair

The MCM card is an electronic product, designed and manufactured to function under somewhat adverse conditions. As with any product, through age, misapplication, or any one of many possible problems, the card may require repair.

When purchased from ProSoft Technology, the module has a one year parts and labor warranty according to the limits specified in the warranty. Replacement and/or returns should be directed to the distributor from whom the product was purchased. If you need to return the card for repair, it is first necessary to obtain an RMA number from ProSoft Technology. Please call the factory for this number and display the number prominently on the outside of the shipping carton used to return the card.

General Warranty Policy

ProSoft Technology, Inc. (Hereinafter referred to as ProSoft) warrants that the Product shall conform to and perform in accordance with published technical specifications and the accompanying written materials, and shall be free of defects in materials and workmanship, for the period of time herein indicated, such warranty period commencing upon receipt of the Product.

This warranty is limited to the repair and/or replacement, at ProSoft's election, of defective or non-conforming Product, and ProSoft shall not be responsible for the failure of the Product to perform specified functions, or any other non-conformance caused by or attributable to: (a) any misapplication of misuse of the Product; (b) failure of Customer to adhere to any of ProSoft's specifications or instructions; (c) neglect of, abuse of, or accident to, the Product; or (d) any associated or complementary equipment or software not furnished by ProSoft.

Limited warranty service may be obtained by delivering the Product to ProSoft and providing proof of purchase or receipt date. Customer agrees to insure the Product or assume the risk of loss or damage in transit, to prepay shipping charges to ProSoft, and to use the original shipping container or equivalent. Contact ProSoft Customer Service for further information.

Limitation of Liability

EXCEPT AS EXPRESSLY PROVIDED HEREIN, PROSOFT MAKES NO WARRANT OF ANY KIND, EXPRESSED OR IMPLIED, WITH RESPECT TO ANY EQUIPMENT, PARTS OR SERVICES PROVIDED PURSUANT TO THIS AGREEMENT, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANT ABILITY AND FITNESS FOR A PARTICULAR PURPOSE. NEITHER PROSOFT OR ITS DEALER SHALL BE LIABLE FOR ANY OTHER DAMAGES, INCLUDING BUT NOT LIMITED TO DIRECT, INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION IN CONTRACT OR TORT (INCLUDING NEGLIGENCE AND STRICT LIABILITY), SUCH AS, BUT NOT LIMITED TO, LOSS OF ANTICIPATED PROFITS OR BENEFITS RESULTING FROM, OR ARISING OUT OF, OR IN CONNECTION WITH THE USE OR FURNISHING OF EQUIPMENT, PARTS OR SERVICES HEREUNDER OR THE PERFORMANCE, USE OR INABILITY TO USE THE SAME, EVEN IF PROSOFT OR ITS DEALER'S TOTAL LIABILITY EXCEED THE PRICE PAID FOR THE PRODUCT.

Where directed by State Law, some of the above exclusions or limitations may not be applicable in some states. This warranty provides specific legal rights; other rights that vary from state to state may also exist. This warranty shall not be applicable to the extent that any provisions of this warranty is prohibited by any Federal, State or Municipal Law that cannot be preempted.

Hardware Product Warranty Details

<u>Warranty Period</u> : ProSoft warranties hardware product for a period of one (1) year.
<u>Warranty Procedure</u> : Upon return of the hardware Product ProSoft will, at its option, repair or replace Product at no additional charge, freight prepaid, except as set forth below. Repair parts and replacement Product will be furnished on an exchange basis and will be either reconditioned or new. All replaced Product and parts become the property of ProSoft. If ProSoft determines that the Product is not under warranty, it will, at the Customer's option, repair the Product using current ProSoft standard rates for parts and labor, and return the Product freight collect.

B Jumper Configurations

Hardware Overview

When purchasing the MDA-4 product, there are two choices. These choices are as follows:

Description	ProSoft Cat Num PLC	SLC
Module provided by ProSoft	3100	3150

When purchasing the module from ProSoft Technology, many of the jumper configurations will have been factory set. When purchasing the firmware from ProSoft Technology and the Allen-Bradley module from another source, particular attention must be paid to hardware configuration.

Module Jumper Configurations

The following section details the available jumper configurations for the 1771 and 1746 platform solutions. As needed, differences between the module based solutions and the firmware based solutions are highlighted.

3100 for the 1771 Platform

Following are the jumper positions for the ProSoft Technology 3100-MDA-4 module:

Jumper	3100
JW1	N/A
JW2	N/A
JW3	N/A
JW4	Not Used
JW5	8 Pt
JW6	Not Used
JW7	Enabled
JW8	As Needed
JW9	As Needed

JW5 Backplane 8/16 point **8 Point**
The module should be operated in the 8 pt mode only.

JW7 Battery Enable / Disable **Enabled**
This jumper should be placed in the Enabled position when the module is powered up. Although not critical to the operation of the module, this will back up some data registers in the module during a power failure or reset.

JW8/9 RS Configuration for Port 1 and 2 **See options on module**
The default from factory is RS-232, but all options are supported by the MDA-4 firmware

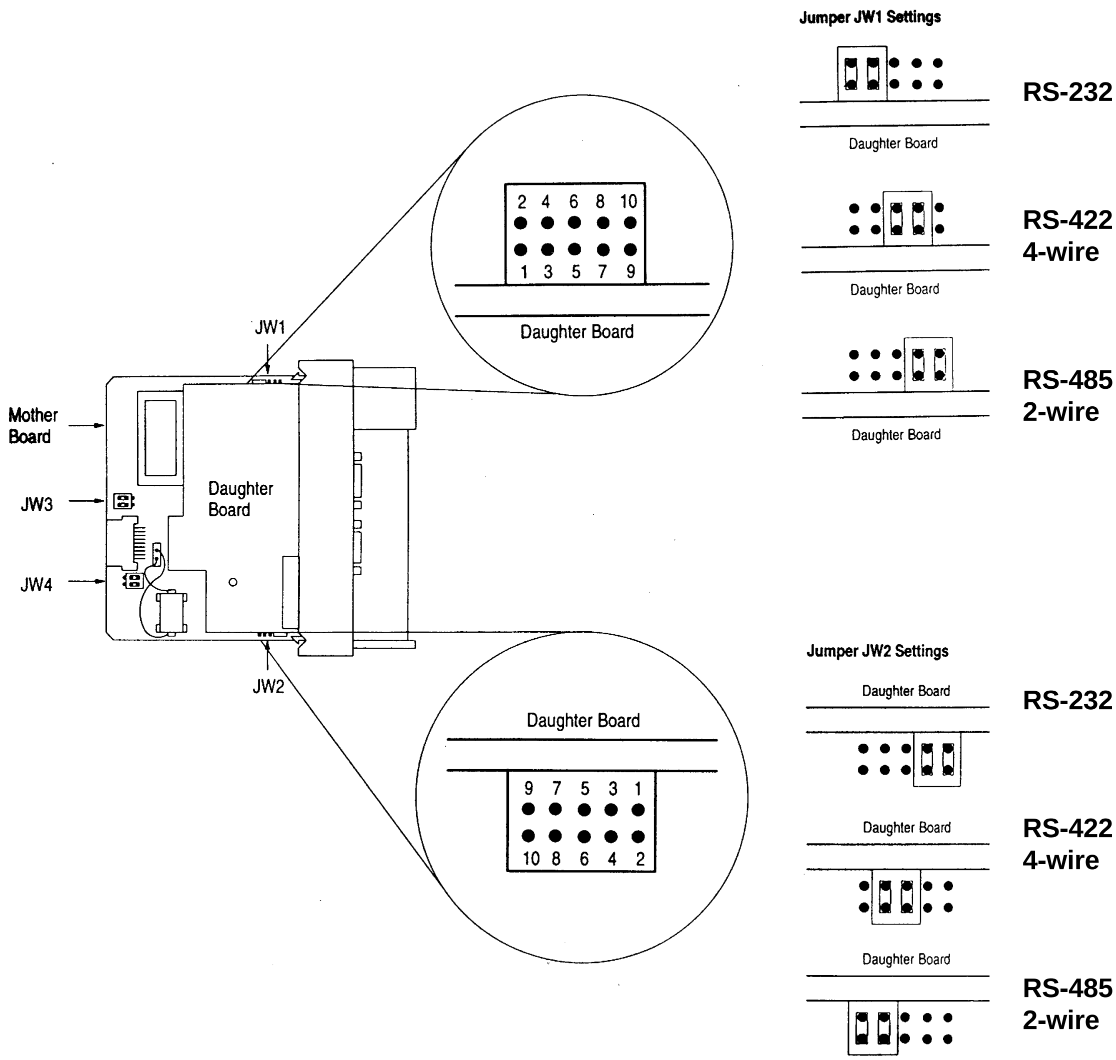
3150 for the 1746 Platform

Following are the jumper positions for the ProSoft Technology 3150-MDA-4 module:

Jumper	3150-MDA-4
JW1	As Needed – See Below
JW2	As Needed – See Below
JW3	N/A
JW4	N/A

JW1/2 RS configuration for port 1 and 2
The default from factory is RS-232, but all options are supported by the MDA-4 firmware.

Communication Port
Jumper Settings for 3150 Modules - JW1 & JW2



C SLC Programming Considerations

The 3150-MDA-4 is also very easy to get operational.

In order to implement the sample logic, the user must make sure that the correct processor and rack size match up. Also, should it be necessary to re-locate the MDA-4 module, the user should be certain to configure the correct slot as a 1746-BAS 5/02 Configuration.

When initially setting up the SLC program file, or when moving the module from one slot to another, the user must configure the slot to accept the MDA-4 module.

> It is important that the slot containing the ProSoft module be configured as follows:
> - 1746-BAS module or enter 13106 for the module ID code
> - Configure the M0/M1 files for 64 words
> - Configure I/O for 8 words

The following is a step by step on how to configure these files using Allen-Bradley APS software. ICOM software users should follow similar steps.

From the Main Menu:

1) Select the correct processor program and F3 for Offline programming
2) F1 for Processor Functions
3) F1 for Change Processor
   Modify the processor here if necessary (Note the MDA-4 will only work with 5/02 or greater processors
4) F5 for Configure I/O
   Select 1746-BAS module for SLC 5/02 or greater, or enter 13106 for module code
5) F9 for SPIO Config when the correct slot is highlighted
6) F5 Advanced Setup
7) F5 for M0 file length - type in 64 and Enter
8) F6 for M1 file length - type in 64 and Enter

Esc out and save configuration

D Example Ladder Logic

Overview

The following ladder logic provides an example for the ladder logic necessary to integrate the 3100-MDA-4 and the 3150-MDA-4 modules into their respective processor platforms. This logic can be incorporated directly as is, or if desired modified as needed for the application.

Data Files

The examples use the same memory map for both of the platforms, with the exception of the actual block transfer data and control files.

The memory map for the example application has been detailed in the attached data table listing.

In this example application, the following configuration and data table layout is used (Note that the application programmer may select any PLC data files (Integer) if the files used in the example are not available):

(Note that the data file listings that are included in this manual do not reflect actual values collected from the CM4 instruments.)

Word	Slave 1	Slave 2	Slave 3
0	Block0	Block6	Block12
	Block1	Block7	Block13
	Block2	Block8	Block14
	Block3	Block9	Block15
	Block4	Block10	Block16
299	Block5	Block11	Block17
	N10	**N11**	**N12**

```
Example PLC Ladder Logic                                              Sat May 31, 1997   Page 1
Program Listing Report                      PLC-5/25    File MDA4                      Rung 2:0

Rung 2:0
       BT READ AND REGISTER TRANSFER FROM MODULE DECODING
       BT READ from module.  This rung of logic is responsible for transferring data
       from the module into the PLC data table.  The paging is controlled by the BTR
       Block ID number(word 0 in Data file).  See manual for complete explanation.
|                                                                       BT READ               |
| BT WRITE  |BT READ                                                    FROM                  |
| ENABLE    |ENABLE                                                     MODULE                |
|   N7:300     N7:400                                         +BTR--------------------+       |
+----]/[--------]/[----------+--------------------------------+BLOCK TRANSFER READ    +-(EN)+-+
|      15         15         |                                |Rack                 00|     | |
|                            |                                |Group                 0+-(DN)| |
|                            |                                |Module                0|     | |
|                            |                                |Control block    N7:400+-(ER)| |
|                            |                                |Data file        N7:410|     | |
|                            |                                |Length               64|     | |
|                            |                                |Continuous            N|     | |
|                            |                                +-----------------------+     | |
|                            |     DECODE               TRANSFER                            | |
|                            |     BT READ              BTR BLOCK                           | |
|                            |     BLOCK ID             ID VALUE                            | |
|                            |+LIM---------------+ +MOV---------------+                     | |
|                            ++LIMIT TEST (CIRC) +++MOVE              ++--------------------+ |
|                            ||Low limit        0|||Source      N7:410||                    | |
|                            ||                  |||                 1||                    | |
|                            ||Test        N7:410|||Destination N7:409||                    | |
|                            ||                 1|||                50||                    | |
|                            ||High limit       5||+------------------+|                    | |
|                            ||                  ||     INTEGER        |                    | |
|                            |+------------------+|     FILE           |                    | |
|                            |                    |     NUMBER         |                    | |
|                            |                    |+MOV---------------+|                    | |
|                            |                    ++MOVE              ++                    | |
|                            |                     |Source          10|                     | |
|                            |                     |                  |                     | |
|                            |                     |Destination N7:408|                     | |
|                            |                     |                 0|                     | |
|                            |                     +------------------+                     | |
|                            |     DECODE               NORMALIZE                           | |
|                            |     BT READ              TO 0-5                              | |
|                            |     BLOCK ID             RANGE                               | |
|                            |+LIM---------------+ +SUB---------------+                     | |
|                            ++LIMIT TEST (CIRC) +++SUBTRACT          ++--------------------+ |
|                            ||Low limit        6|||Source A    N7:410||                    | |
|                            ||                  |||                 1||                    | |
|                            ||Test        N7:410|||Source B         6||                    | |
|                            ||                 1|||                  ||                    | |
|                            ||High limit      11|||Destination N7:409||                    | |
|                            ||                  |||                50||                    | |
|                            |+------------------+|+------------------+|                    | |
|                            |                    |     INTEGER        |                    | |
|                            |                    |     FILE           |                    | |
|                            |                    |     NUMBER         |                    | |
|                            |                    |+MOV---------------+|                    | |
|                            |                    ++MOVE              ++                    | |
|                            |                     |Source          11|                     | |
|                            |                     |                  |                     | |
|                            |                     |Destination N7:408|                     | |
|                            |                     |                 0|                     | |
|                            |                     +------------------+                     | |
|                           vvv                                                            vvv|
```

```
Example PLC Ladder Logic                                              Sat May 31, 1997   Page 2
Program Listing Report                      PLC-5/25    File MDA4                      Rung 2:0
|                           ^^^                                                            ^^^|
|                            |     DECODE               NORMALIZE                           | |
|                            |     BT READ              TO 0-5                              | |
|                            |     BLOCK ID             RANGE                               | |
|                            |+LIM---------------+ +SUB---------------+                     | |
|                            ++LIMIT TEST (CIRC) +++SUBTRACT         ++--------------------+ |
|                            ||Low limit       12|||Source A    N7:410||                    | |
|                            ||                  |||                 1||                    | |
|                            ||Test        N7:410|||Source B        12||                    | |
|                            ||                 1|||                  ||                    | |
|                            ||High limit      17|||Destination N7:409||                    | |
|                            ||                  |||                50||                    | |
|                            |+------------------+|+------------------+|                    | |
|                            |                    |     INTEGER        |                    | |
|                            |                    |     FILE           |                    | |
|                            |                    |     NUMBER         |                    | |
|                            |                    |+MOV---------------+|                    | |
|                            |                    ++MOVE              ++                    | |
|                            |                     |Source          12|                     | |
|                            |                     |                  |                     | |
|                            |                     |Destination N7:408|                     | |
|                            |                     |                 0|                     | |
|                            |                     +------------------+                     | |
|                            |     DECODE                                   CALC            | |
|                            |     BT READ                                  OFFSET          | |
|                            |     BLOCK ID                                 POINTER         | |
|                            |+LIM---------------+                    +CPT---------------+ | |
|                            ++LIMIT TEST (CIRC) ++--------------------+COMPUTE           +++ |
|                            ||Low limit        0||                    |Destination N7:409||| |
|                            ||                  ||                    |                50||| |
|                            ||Test        N7:410||                    |Expression        ||| |
|                            ||                 1||                    |N7:409 * 50       ||| |
|                            ||High limit      60||                    +------------------+|| |
|                            ||                  ||          PERFORM                        || |
|                            |+------------------+|          TRANSFER                       || |
|                            |                    |          TO DATA                        || |
|                            |                    |          FILE                           || |
|                            |                    |+FAL------------------------------+     || |
|                            |                    ++FILE ARITH/LOGICAL              +-(EN)+| |
|                            |                     |Control                     R6:0|      | |
|                            |                     |Length                        50+-(DN) | |
|                            |                     |Position                       0|      | |
|                            |                     |Mode                         ALL+-(ER) | |
|                            |                     |Destination    #N[N7:408][N7:409]|      | |
|                            |                     |                                0|      | |
|                            |                     |Expression                       |      | |
|                            |                     |#N7:412                          |      | |
|                            |                     +---------------------------------+      | |
|                            |                                               ENCODES        | |
|                            |                                               BT WRITE       | |
|                            |                                               BLOCK ID       | |
|                            |                                          +MOV---------------+| |
|                            +------------------------------------------+MOVE              ++ |
|                            |                                          |Source      N7:411|| |
|                            |                                          |                 1|| |
|                            |                                          |Destination N7:310|| |
|                            |                                          |                 0|| |
|                            |                                          +------------------+| |
|                            |USER CFG                                       ENCODES        | |
|                            |DOWNLOAD                                       BT WRITE       | |
|                            |SELECT                                         BLOCK ID       | |
|                            |    B3                                    +MOV---------------+| |
|                            +---] [------------------------------------+MOVE              ++ |
|                                  0                                     |Source         255|  |
|                                                                        |                  |  |
|                                                                        |Destination N7:310|  |
|                                                                        |                 0|  |
|                                                                        +------------------+  |
```

```
Example PLC Ladder Logic                                                  Sat May 31, 1997   Page 3
Rung 2:1
       This rung calls a subroutine that toggles the Enable bits on the writes based
       on the done bit being received.
| N7:400                                                                +JSR---------------+ |
+--] [------------------------------------------------------------------+JUMP TO SUBROUTINE+-+
|    13                                                                 |Prog file number 3| |
|                                                                       |Input parameter   | |
|                                                                       |Return parameter  | |
|                                                                       +------------------+ |

Rung 2:2
       WRITES DATA,COMMAND LIST OR CONFIGURATION BLOCK TO MODULE
       This rung is responsible for transferring data to the module.  This data
       includes the command blocks (0 and 1) and the configuration block (255).
|                                                  DECODE                 WRITE TO           |
| BT READ   |BT WRITE                              BT WRITE               BT WRITE           |
| ENABLE    |ENABLE                                BLOCK                  BUFFER             |
|   N7:400     N7:300                             +EQU---------------+  +COP--------------------+  |
+----]/[--------]/[-------------------------++EQUAL              +--+COPY FILE             ++-+
|      15         15                          ||Source A    N7:310|  |Source           #N9:0|| |
|                                             ||                  0|  |Destination     #N7:311|| |
|                                             ||Source B          0|  |Length              50|| |
|                                             ||                   |  +----------------------+| |
|                                             |+------------------+                          | |
|                                             |     DECODE                 WRITE TO           | |
|                                             |     BT WRITE               BT WRITE           | |
|                                             |     BLOCK                  BUFFER             | |
|                                             |+EQU---------------+  +COP--------------------+| |
|                                             ++EQUAL              +--+COPY FILE             ++ |
|                                             ||Source A    N7:310|  |Source          #N9:50|| |
|                                             ||                  0|  |Destination     #N7:311|| |
|                                             ||Source B          1|  |Length              50|| |
|                                             ||                   |  +----------------------+| |
|                                             |+------------------+                          | |
|                                             |     DECODE                WRITE TO            | |
|                                             |     BT WRITE              BT WRITE            | |
|                                             |     BLOCK                 BUFFER              | |
|                                             |+EQU---------------+ +COP--------------------+ | |
|                                             ++EQUAL              +++COPY FILE             +++ |
|                                             ||Source A    N7:310|||Source           #N7:0||| |
|                                             ||                  0|||Destination     #N7:311||| |
|                                             ||Source B        255|||Length              30||| |
|                                             ||                   ||+----------------------+|| |
|                                             |+------------------+|             USER CFG  || |
|                                             |                    |             DOWNLOAD  || |
|                                             |                    |             SELECT    || |
|                                             |                    |                 B3    || |
|                                             |                    +------------------(U)----+| |
|                                             |                                         0      | |
|                                             |                         BT WRITE                | |
|                                             |                         TO MODULE               | |
|                                             |                 +BTW--------------------+      | |
|                                             +-----------------+BLOCK TRANSFER WRITE   +-(EN)+ |
|                                                               |Rack                 00|       |
|                                                               |Group                 0+-(DN)  |
|                                                               |Module                0|       |
|                                                               |Control block    N7:300+-(ER)  |
|                                                               |Data file        N7:310|       |
|                                                               |Length               64|       |
|                                                               |Continuous            N|       |
|                                                               +-----------------------+       |
Rung 2:3
|                                                                                           |
+-------------------------------------------------[END OF FILE]---------------------------------+
|                                                                                           |

Rung 3:0
|                                                                                           |
+-------------------------------------------------[END OF FILE]---------------------------------+
|                                                                                           |
```

Example PLC Ladder Logic Sat May 31, 1997
Data Table Report PLC-5/25 File MDA4 Data Table

File N7:0

Address	0	1	2	3	4	5	6	7	8	9
N7:0	5	5	1	3	0	0	0	0	0	0
N7:10	257	257	257	0	0	0	0	0	0	0
N7:20	0	0	0	0	0	0	0	0	0	0
N7:30	0	0	0	0	0	0	0	0	0	0

File N9:0

Address	0	1	2	3	4	5	6	7	8	9
N9:0	63	0	31	0	768	0	0	0	0	0
N9:10	63	0	0	0	0	0	0	0	0	0
N9:20	63	0	0	0	0	0	0	0	0	0
N9:30	0	0	0	0	0	0	0	0	0	0
N9:40	0	0	0	0	0	0	0	0	0	0
N9:50	0	0	0	0	0	0	0	0	0	0
N9:60	0	0	0	0	0	0	0	0	0	0
N9:70	0	0	0	0	0	0	0	0	0	0
N9:80	0	0	0	0	0	0	0	0	0	0
N9:90	0	0	0	0	0	0	0	0	0	

File N10:0

Address	0	1	2	3	4	5	6	7	8	9
N10:0	3864	0	0	0	0	0	1	0	0	0
N10:10	6	2	1997	6	40	28	0	0	0	0
N10:20	1211	516	-1	-32296	-16334	0	24257	-1	0	0
N10:30	-1	-1	152	121	132	135	0	0	0	0
N10:40	20040	13101	18761	129	153	8898	2624	8898	3356	0
N10:50	0	0	0	0	0	0	0	0	0	0
N10:60	20040	13101	18761	129	122	8898	2624	8898	3356	0
N10:70	0	0	0	0	0	0	0	0	0	0
N10:80	20040	13101	18761	129	131	8898	2624	8898	3356	0
N10:90	0	0	0	0	0	0	0	0	0	0
N10:100	20040	13101	18761	129	135	8898	2624	8898	3356	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
N10:210	0	0	0	0	0	0	0	0	0	0
N10:220	0	0	0	0	0	0	0	0	0	0
N10:230	0	0	0	0	0	0	0	0	0	0
N10:240	0	0	0	0	0	0	0	0	0	0
N10:250	0	0	0	0	0	0	0	0	0	0
N10:260	0	0	0	0	0	0	0	0	0	0
N10:270	0	0	0	0	0	0	0	0	0	0
N10:280	0	0	0	0	0	0	0	0	0	0
N10:290	19780	16692	12590	12336	12594	12338	0	0	0	0

File N11:0

Address	0	1	2	3	4	5	6	7	8	9
N11:0	1751	0	0	12	0	0	0	0	0	0
N11:10	6	2	1997	6	40	29	0	0	0	0
N11:20	1211	516	-1	-32296	-16334	0	24257	-1	0	0
N11:30	-1	-1	153	121	132	134	0	0	0	0
N11:40	20040	13101	18761	129	152	8898	2624	8898	3357	0
N11:50	0	0	0	0	0	0	0	0	0	0
N11:60	20040	13101	18761	129	121	8898	2624	8898	3357	0
N11:70	0	0	0	0	0	0	0	0	0	0
N11:80	20040	13101	18761	129	131	8898	2624	8898	3354	0
N11:90	0	0	0	0	0	0	0	0	0	0
N11:100	20040	13101	18761	129	135	8898	2624	8898	3354	0
N11:110	0	0	0	0	0	0	0	0	0	0
N11:120	0	0	0	0	0	0	0	0	0	0
N11:130	0	0	0	0	0	0	0	0	0	0
N11:140	0	0	0	0	0	0	0	0	0	0
N11:150	0	0	0	0	0	0	0	0	0	0
N11:160	0	0	0	0	0	0	0	0	0	0
N11:170	0	0	0	0	0	0	0	0	0	0
N11:180	0	0	0	0	0	0	0	0	0	0
N11:190	0	0	0	0	0	0	0	0	0	0
N11:200	0	0	0	0	0	0	0	0	0	0
N11:210	0	0	0	0	0	0	0	0	0	0
N11:220	0	0	0	0	0	0	0	0	0	0
N11:230	0	0	0	0	0	0	0	0	0	0
N11:240	0	0	0	0	0	0	0	0	0	0
N11:250	0	0	0	0	0	0	0	0	0	0
N11:260	0	0	0	0	0	0	0	0	0	0
N11:270	0	0	0	0	0	0	0	0	0	0
N11:280	0	0	0	0	0	0	0	0	0	0
N11:290	0	0	0	0	0	0	0	0	0	0

File N12:0

Address	0	1	2	3	4	5	6	7	8	9
N12:0	1740	0	0	0	0	0	0	0	0	0
N12:10	6	2	1997	6	40	27	0	0	0	0
N12:20	1211	516	-1	-32296	-16334	0	24257	-1	0	0
N12:30	-1	-1	152	123	131	135	0	0	0	0
N12:40	20040	13101	18761	129	154	8898	2624	8898	3355	0
N12:50	0	0	0	0	0	0	0	0	0	0
N12:60	20040	13101	18761	129	122	8898	2624	8898	3355	0
N12:70	0	0	0	0	0	0	0	0	0	0
N12:80	20040	13101	18761	129	131	8898	2624	8898	3355	0
N12:90	0	0	0	0	0	0	0	0	0	0
N12:100	20040	13101	18761	129	133	8898	2624	8898	3355	0
N12:110	0	0	0	0	0	0	0	0	0	0
N12:120	0	0	0	0	0	0	0	0	0	0
N12:130	0	0	0	0	0	0	0	0	0	0
N12:140	0	0	0	0	0	0	0	0	0	0
N12:150	0	0	0	0	0	0	0	0	0	0
N12:160	0	0	0	0	0	0	0	0	0	0
N12:170	0	0	0	0	0	0	0	0	0	0
N12:180	0	0	0	0	0	0	0	0	0	0
N12:190	0	0	0	0	0	0	0	0	0	0
N12:200	0	0	0	0	0	0	0	0	0	0
N12:210	0	0	0	0	0	0	0	0	0	0
N12:220	0	0	0	0	0	0	0	0	0	0
N12:230	0	0	0	0	0	0	0	0	0	0
N12:240	0	0	0	0	0	0	0	0	0	0
N12:250	0	0	0	0	0	0	0	0	0	0
N12:260	0	0	0	0	0	0	0	0	0	0
N12:270	0	0	0	0	0	0	0	0	0	0
N12:280	0	0	0	0	0	0	0	0	0	0
N12:290	0	0	0	0	0	0	0	0	0	0