

DMC-1000™

DC Motor Drive Control

Operation / Installation Manual

Manual Part Number: S8M5004

Date: August 11, 1998



TABLE OF CONTENT

SECTION 1 - OVERVIEW	1
1.0 Functional Description	1
1.1 Remote I/O Control	1
1.2 Motor Output	1
1.3 Operational Status	1
SECTION 2 - INSTALLATION	3
2.0 Enclosure Installation	3
2.1 Dc Motor Connector Pin-Out	4
2.2 Optional Encoder Connections	4
2.3 Enable Optional Encoder	5
2.4 Motor Connection	5
2.5 Remote I/O Connector Pin-Out	6
SECTION 3 - OPERATION	7
3.0 Theory of Operation	7
3.1 Control Calibration	7
SECTION 4 – RS-232 TERMINAL PROTOCOL	9
4.1 Selecting RS-232 or LAN Operation	9
4.2 RS-232 Protocol	9
4.3 RS-232 Command Summary	10
4.4 Special Command Codes	11
4.5 LAN Command Protocol	11
4.8 LAN Command Summary	12
SECTION 5 - DMC-1000 Programmable Sequence Protocol	13
5.1 PLC Command Protocol	13
5.2 Relay Outputs	13

5.3 PLC Command Summary	14
SECTION 6 – DRAWINGS AND PARTS LISTS	19
6.1 System Configuration	19
6.2 System Schematic Diagram	20
6.3 DMC-1000 Enclosure Details	21

SECTION 1 - OVERVIEW

1.0 Functional Description

The DMC-1000™ DC Motor Controller is a SCR motor speed controller using an embedded micro controller to provide motor speed regulation and torque compensation. The controller can be used to control permanent magnet (PM) or shunt wound field motors up to ¼ hp and is designed to be used with the WSC-1000™ Weld Sequence Controller or user supplied controllers.

The DMC-1000™ is used to control linear DC drive motors for wire feed and or travel speed functions. The DMC-1000™ provides speed and torque regulation using an optical encoder or phase sampled Back EMF. The embedded controller provides precise motor braking and anti-plugging features to extend motor life.

The DMC-1000™ includes a LAN CPU controller that interfaces the SCR motor drive to the WSC-1000™ Local Area Network (LAN) communications port. This allows the WSC-1000™ to control all motor function via the LAN serial port.

1.1 Remote I/O Control

The DMC-1000™ provides remote control functions using the PLC controller. The Remote I/O has four (4) 24vdc inputs and two (2) relay outputs that can be user configured. A remote I/O connector located on the rear of the enclosure provides all user control connections.

1.2 Motor Output

The DMC-1000™ can be used with 90 VDC permanent magnet or shunt field wound motors up to ¼ H.P. A motor connector located on the rear of the enclosure provides all user motor connections. The control can be used with or without an optical tachometer. The control uses a 60-line optical tach input and can provide a 5 VDC or 15VDC output for the tachometer. The Tachometer input mode is enabled by an internal jumper connection.

1.3 Operational Status

The operational status of the DMC-1000™ is displayed using 4 LED's located on the front panel. The **POWER +24V** LED indicates that power is applied to the controller. The **BRAKE** LED indicates when the motor is at rest or is braking to a stop condition. The **DRIVE ON** led indicates when the drive is active. The **REVERSE** LED indicates when the motor is operating in the reverse direction. Four Communication LED's indicate the current serial communications status. COM R/T indicate RS-232 status and LAN R/T indicate WSC-1000 LAN communications.

Note: When the motor is active in the forward direction the **DRIVE ON** LED will illuminate. When the motor is active in the reverse direction, the **DRIVE ON** and **REVERSE LED** will illuminate.

1.4 Control Specification

The following are the electrical specifications for the DMC-1000™:

- Power Input – 120 vac \pm 10% @ 5amps
- Armature Current - 0.5 – 3.0 amps
- Armature Voltage - 0 – 100 vdc
- Field Voltage - 110 vdc nominal
- Field Current - 2.0 amps maximum
- Encoder Input - 5 or 15 vdc 60 lines/rpm max frequency 3.0khz
- Encoder output - 24 vdc pulse
- Inputs (INP 1 - INP 2) – 24 vdc @ 20ma (Active High)
- Outputs (CR1 – CR2) – Dry Relay contact 110 VAC @ 0.1 amp
- Limit Inputs – 24 vdc @ 20ma (Active High)

The following are the mechanical specifications for the DMC-1000™ control:

- Dimensions – 2.0"H x 8.5"W x 11"L (51mm x 165mm x 280mm)
- Mounting Dimensions – 7.5"W x 8.5"L (4 ea 10-32 tapped hole)
- Weight – 5 lbs (2.27 kg)
- Operating Temperature - -10°F to +140°F (-23°C to +60°C)

SECTION 2 - INSTALLATION

2.0 Enclosure Installation

Locate the DMC-1000™ enclosure in a convent location. The enclosure can be installed using the four 10-32 mounting holes located on the bottom of the enclosure. Refer to Figure 2-1 for mounting dimensions:

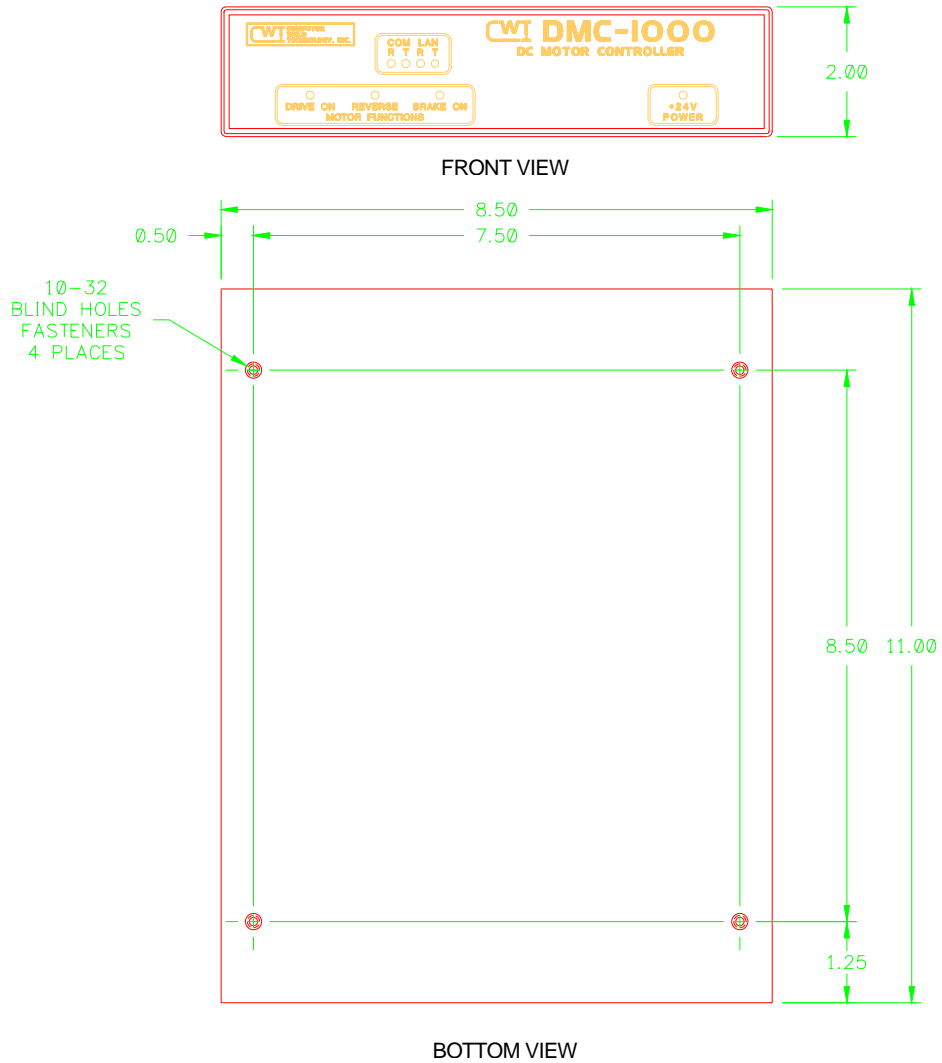


Fig 2-1 DMC-1000 Enclosure Mounting dimension

Connect power cable S3W5043 to suitable 115 VAC power outlet. Connect motor control cable to MOTOR connector on rear of enclosure. Connect user supplied I/O cable to I/O connector on rear of enclosure.

2.1 Dc Motor Connector Pin-Out

Several cables are available from the factory, which and can be used for connecting a DC drive motor to the DMC-1000™. Consult the factory for available cable assemblies. The following is the pin-out for the DMC-1000 motor connector:

PIN	DESCRIPTION
A	+15 VDC for optional encoder
B	+5 VDC for optional encoder
C	Limit switch input common
D	CW Limit switch input
E	CCW Limit switch input
F	Chassis Ground
G	Encoder Pulse input (TTL or 15 Volt Pulse)
H	Encoder Common
J	Motor Field -
K	Motor Armature -
L	Motor Field +
M	Motor Armature +

Figure 2-2 Motor Connector Pin-out

2.2 Optional Encoder Connections

The DMC-1000™ can use a 5-volt or 15 volt encoder. Both power outputs are provided on the connector. Connect the optional motor encoder to the correct voltage output. Use Pin A for a 15-volt encoder or Pin B for a 5-volt encoder. The DMC-1000™ is designed for a 60-line encoder and can be used for motor speeds up to 3000 rpm. For faster speeds, contact the factory. The following is a typical connection for a 5-volt TTL encoder:

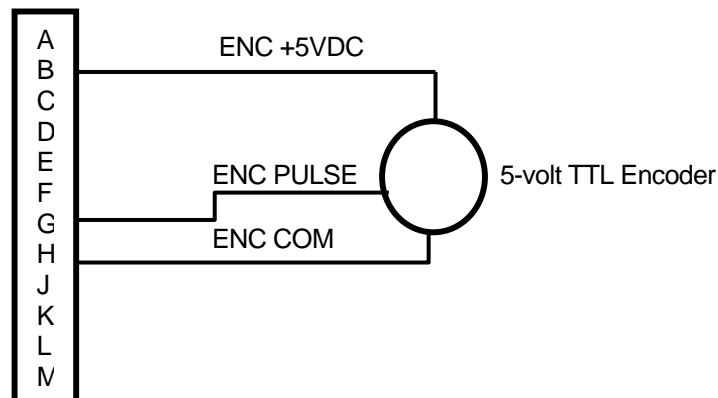


Fig 2-3 5-volt TTL encoder connection

The following is the connection diagram for a 15 VDC encoder:

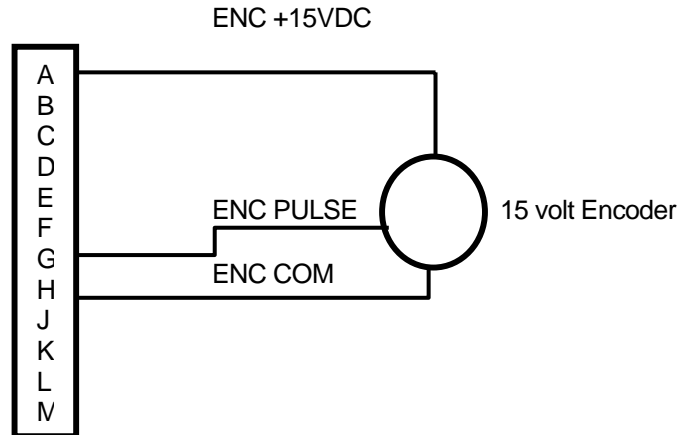


Fig 2-4 15-volt TTL encoder connection

2.3 Enable Optional Encoder

When using an optional encoder the DMC-1000™ must be configured for encoder input. To select the Encoder mode remove the cover from the DMC enclosure by removing the six (6) screws on the side of the cover. Locate the jumper labeled **JP-3 TACH** and install a jumper. This enables the Tachometer mode of operation.

Note: For non-tach applications the jumper on **JP-3 TACH** must be removed.

2.4 Motor Connection

The DMC-1000 can be used with a permanent magnet, or shunt field motors with an armature rating of 90 – 100 volt dc. The following is the connection diagram for a permanent magnet motor:

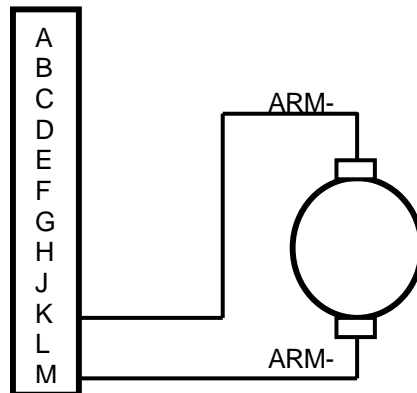


Fig 2-4 Permanent magnetic motor wiring

The following is the connection diagram for a shunt field motor:

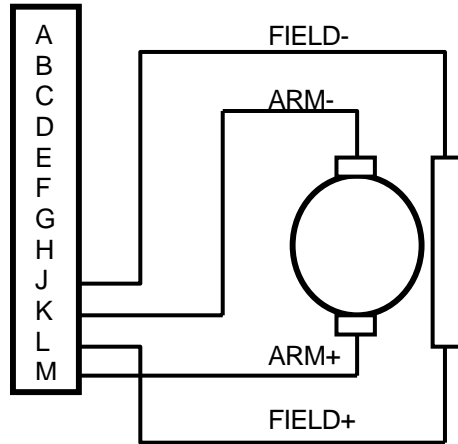


Fig 2-5 Permanent magnetic motor wiring

2.5 Remote I/O Connector Pin-Out

The DMC-1000™ is controlled via the REMOTE I/O connector. The control has two (2) relay outputs and three (3) 24-vdc inputs. The LAN CPU PLC is used to program and control these inputs. The following is the pin-out for the REMOTE I/O connector:

PIN	FUNCTIONAL DESCRIPTION
A	INPUT 1 - 24 vdc Input @ 10ma
B	INPUT 2 – 24 vdc input @ 10ma
C	INPUT 3 – 24 vdc input @ 10ma
D	Input common
E	CR1-A N.O.
F	CR1-B common
G	CR2-A N.O.
H	CR2-B Common

Fig 2-6 Remote I/O connector pin-out

SECTION 3 - OPERATION

3.0 Theory of Operation

The DMC-1000 is a direct off-line SCR motor speed controller. The control uses an embedded micro-controller to provide line synchronization, phased back EMF sampling and direction control logic with anti-plug motor reversing. The phased EMF sample provides precise back EMF sample for improved motor speed regulation. The Anti-plug feature prevents reversing the drive before the motor armature has stopped. This prevents excessive motor current and brush arcing in the motor.

The Micro-Controller synchronizes all input commands to the incoming line frequency and assures proper phase angle firing of the power SCR devices. The Analog speed signal is coupled to a Voltage-to-Frequency convert and is electrically isolated from the off-line control. The remote direction control inputs are optically coupled to the controller. An isolated 24-vdc-power supply is provided for all remote input control functions. The user can use this supply to operate remote input relays or switches for direction control. The supply has a solid-state circuit breaker, which protects the 24-volt I/O control from external shorts. The circuit breaker will reset when the 115-vac power is cycled off.

Optional optical tach input is provided which can be used with a 5 or 15 vdc 60-line encoder. The encoder input is also provided as an output on the remote I/O connector. This output is an isolated 24-vdc pulse representing the encoder input. When an encoder is installed on the motor shaft, the DMC-1000 can be set for encoder feedback by installing a jumper on JP3. When the jumper is installed, the controller uses the tachometer to regulate motor speed. When the jumper is removed, the controller will use the motor back EMF for speed regulation.

3.1 Control Calibration

The DMC-1000™ controller is factory calibrated to produce 2300 rpm with an input of 10 volts. This is performed using a 60 line Tach and a permanent magnet 1/8 motor. No adjustments are required to operate different drive motors. However, there may be a need to check or recalibrate the DMC-1000™ for use with a different motor or motor speed range. To calibrate the DMC-1000™ perform the following steps:

Warning - The DMC-1000™ is a direct off-line controller. Due not used grounded instruments during calibration. 115 vac is exposed on the heat sinks used on the PC Board. Only qualified personal should perform test or calibrations on the control.

- 1) Remove the cover from the DMC-1000™ controller. Connect a calibrated digital voltmeter to the speed reference signal on JP1-3 (-) and JP1-4 (+).
- 2) If encoder is installed on the drive motor connect a frequency counter to the encoder output JP1-7 (+) and JP1-2 (-). A 60-line tachometer will produce a direct readout of motor RPM. If a tach is not installed, use a hand tach to measure actual motor RPM.
- 3) Connect the DMC-1000 to a suitable 115vac source. Make sure that the motor is free to rotate then power up the DMC control.

- 4) Set the speed reference input to zero and activate the drive forward input command. Adjust R4 (MIN) for zero motor RPM.
- 5) Set the speed reference to 10.0 volts and adjust R3 (MAX) for 2300 (HZ or RPM) or desired motor speed. When using a Tachometer the maximum motor speed is 3000 RPM. If the Tachometer input is disabled (Jumper on JP3) the maximum armature voltage is 105 vdc.
- 6) Adjusting R3 may change the motor zero input. Repeat Steps 4 and 5 until desired results are obtained.
- 7) The motor current limit R61 is factory set and should not be changed.
- 8) Disable the forward direction input. Turn the power off and reinstall the DMC cover.

SECTION 4 – RS-232 TERMINAL PROTOCOL

4.1 Selecting RS-232 or LAN Operation

The DMC-1000 is designed to be used with the Computer Weld Technology's WSC-1000 controller using a serial LAN communication or it can be configured for use through a standard RS-232-C serial port. Only one mode of communications can be enabled. The Mode of communications is determined by the S2 dipswitch located on the Drive CPU P.C. Board. Setting all of the switches to the "OPEN" or "OFF" position will enable the RS-232 serial communications. To enable the LAN communications routines the DIP switches must be set to the LAN address for the desired motor control functions. The usable LAN address range is 5 - 8. The following table shows the proper DIP switch settings for LAN communications.

LAN ID FUNCTION	SW1	SW2	SW3	SW4	WSC-1000 CONTROL
5	On	Off	On	Off	Wire Feed Drive
6	Off	On	On	Off	Travel Drive
7	On	On	On	Off	PLC addressable
8	Off	Off	Off	On	PLC addressable

4.2 RS-232 Protocol

The RS-232 communications port is used to off-line program the DMC -1000 sequence and drive parameters when not configured for LAN operations. It is also used to configure the operating parameters for the DMC. The Protocol is a simple ASCII command string that allows the user to upload or down load the various parameters. The serial port is configured for the follow data format:

Baud Rate: 9600, Full Duplex
Word Length: 8 Data Bits, One Stop and no parity
Hand Shaking: None

The Protocol consists of a command string and optional data bytes. The command string is an Alpha character an option number followed by a "=" or "?" followed by optional data and terminated with an ASCII "cr" (0dh) . The "=" will indicate that data is being sent to the select parameter by the host controller. The "?" will indicate a request for data from the DMC to the Host controller. If the host is up loading data to the DMC the data will be placed after the "=" character and will be an ASCII string terminated with an ASCII "cr" (0dh). The following is an example of sending a new drive motor speed to the DMC-1000:

V1=1000(cr) - Sent from Host

To read the DMC drive motor speed send the following command:

V1?(cr) - Sent from Host

1000(cr) - Received from DMC

4.3 RS-232 Command Summary

The following is a summary of the RS-232 serial Commands supported by the DMC-1000:

COMMAND	DESCRIPTION
---------	-------------

A1 - A4	Read DMC-1000 Analog Inputs. The following is a summary of the analog command functions:
----------------	------------------------------------------------------------------------------------------

A1	- DMC 24 vdc power supply voltage (240 = 24.0 vdc)
A2	- Analog input 2 (max value 255)
A3	- Analog input 3 (max value 255)
A4	- Analog input 4 (max value 255)

M0 - M5	Set system control mode functions:
----------------	------------------------------------

M0	- Set programmable sequence number to value.
----	----------------------------------------------

M1	- Read Switch input status (1 = on, 0 = off) BIT 0 = Switch INP1 (value= 1) BIT 1 = Switch INP2 (value= 2) BIT 2 = Switch INP3 (value= 4)
----	------------------------------------------------------------------------------------------------------------------------------------------------------

M2	- Set output relays CR1 - CR2 (1=on, 0=off).
----	----------------------------------------------

BIT 6	= Relay CR1 (value= 64)
BIT 7	= Relay CR2 (value= 128)

M3	- Enable DMC auto sequence routines.
----	--------------------------------------

Command	Function
M3=1	Enable Auto sequence input
M3=2	Clear Auto sequence input
M3=3	Set Initialize sequence input
M3=4	Calculate DAC output scaling for drive motor. Calculate DAC scaling for motor speed in RPM.
M3=5	Calculate maximum position based on external limit switch inputs.

S1 - S75 Programmable Sequence Commands - See Section 5.0 for description of programmable sequence commands and data format.

V1 - V9 Write/Read Welding variables:

V1 = Motor speed.
V2 = Current Drive Position (Scaled encoder count).
V3 = Drive Acceleration Value.
V4 = Encoder scale factor (Counts/distance or Counts/Rev).
V5 = Move to position (Value = New position).
V6 = Maximum Position Limit.
V7 = DAC scaling parameter calculated from M3=4 command (BITS/RPM).
V8 = Encoder counts per revolution of motor (Default 60 counts/rev).
V9 = Drive motor output DAC value.
V10 = PLC Result register 0
V11 = PLC Register 1
V12 = PLC Register 2
V13 = PLC Register 3
V14 = PLC Register 4
V15 = PLC Register 5

4.4 Special Command Codes

In addition to the terminal commands the DMC-1000 supports several special control key functions. These functions are used to save the programmed data in the EEPROM and to clear any pending terminal commands. The following commands are generated by pressing the CTRL" and specified letter key at the same time. When sending any of the following control codes, the DMC-1000 will respond with an ASCII "CR". The following is a summary of the special control character function supported by the DMC-1000:

Control Command Function Code

^W Save current parameters and sequence values to DMC-1000 EEPROM
^C Reset the terminal serial port and clear any pending terminal commands.
^S Up Load stored sequence commands from DMC to terminal.
^R Load sequence commands from EEPROM to RAM

4.5 LAN Command Protocol

The above commands are only available when the RS-232 serial communications is enabled. If the DMC-1000 is configured for LAN communications then the following commands maybe used to configure the controller using the WSC-1000 terminal port and a P.C. based communications program. To send DMC drive commands use the L#= WSC terminal command where # is the axis drive LAN ID number. The string following the "=" will be sent to the selected axis drive.

Example: L6=V=1000 (cr) Set travel speed to 1000 RPM.

4.8 LAN Command Summary

The following is a summary of the DMC-1000 LAN drive commands:

^S	Save current drive configuration to drive EEPROM and use as default.
^L	Load default drive EEPROM data.
^Q	Terminate current drive move command.
M=	Move to position specified by value.
A=	Set output current to value specified.
V=	Set velocity to value specified.
H	Reset drive and move to Home position.
O	Perform selected command options:
	Command Function
O=1	Enable Auto sequence input
O=2	Clear Auto sequence input
O=3	Set Initialize "Home" sequence input
O=4	Calculate DAC output scaling for drive motor in RPM.
O=5	Calculate maximum position based on external limit switch inputs.
O=6	Reset PLC
I	Inhibit axis drive output power.
S	Set auto sequence parameter S1 - S50 to specified move, velocity or delay parameter. EXAMPLE: <i>S1=V1000, set sequence 1 to set motor speed equal to 1000 rpm.</i>
G	Start axis drive auto PLC sequence routine.
Q	Quit axis drive auto PLC sequence routine.
Z	Clear all axis drive PLC sequence parameters.
E	Enable axis drive power output.
C=	Encoder Counts/revolutions (Default = 60)
R=	Jog drive position forward by specified value.
L=	Jog drive position reverse by specified value.
U=	Set axis drive acceleration constant to value specified.
T	Calculate maximum position based on external limit switch inputs.
F=	Set Encoder scale factor (Counts/Distance).
+=	Drive motor in the forward direction at speed specified by the value.
-=	Drive motor in the reverse direction at speed specified by the value.

SECTION 5 - DMC-1000 Programmable Sequence Protocol

5.1 PLC Command Protocol

The DMC programmable sequence consists of a 3 byte command. The First byte is the command byte followed by a two-byte value:

{Byte1}, {Byte2 (MSB):}Byte3 (LSB)}

The value bytes must be set even if not required by the command. The value bytes may be branch addresses or real data passed to the selected function. A comma must separate the Command and value. The MSB byte and LSB byte of the value maybe separated by a colon (":"). The colon will cause the MSB and LSB byte to concatenate to form a single two-byte value.

When setting MSB and LSB bytes for specific command function the MSB and LSB byte may be separated by a ":" or the total value may be specified by the decimal equivalent of the two bytes. To set the decimal value for the MSB and LSB bytes use the following equation:

$$\text{Decimal Value} = (\text{MSB} \times 256) + \text{LSB}$$

The following is an example of how to set sequence 4 to the "SWITCH ON" command (1) and branch to sequence number 10 when "OFF" (MSB) and to test switch input 1 (LSB):

$$\text{Decimal Value} = (\text{MSB} \times 256) + \text{LSB} = (10 \times 256) + 1 = 2561$$

Decimal Command sent to DMC: **S4=1,2561**

Optional Command format: **S4=1,10:1**

5.2 Relay Outputs

When Setting / Resetting the relay outputs the individual relays are selected by setting the corresponding data bits in the LSB byte. To set multiply relays with the same command add the decimal value for each relay and use the result as the value for the command. The following is an example of programming sequence 5 to set relay CR1 and CR2 using the "SET RELAY" command (3):

$$\text{Decimal Value for CR1} = 64 \text{ and } \text{CR6} = 128$$

$$\text{Decimal Value} = 64 + 128 = 192$$

Command sent DMC: **S5=8,192**

The following is the decimal value for individual bits used for the relay outputs and switch inputs:

<i>BIT NUMBER</i>	<i>DECIMAL</i>	<i>RELAY SWITCH</i>	<i>INPUT SWITCH</i>
<i>BIT 0</i>	<i>1</i>	<i>N/A</i>	<i>INP1</i>
<i>BIT 1</i>	<i>2</i>	<i>N/A</i>	<i>INP2</i>
<i>BIT 2</i>	<i>4</i>	<i>N/A</i>	<i>INP3</i>
<i>BIT 3</i>	<i>8</i>	<i>N/A</i>	<i>INP4</i>
<i>BIT 4</i>	<i>16</i>	<i>N/A</i>	<i>CW Limit</i>
<i>BIT 5</i>	<i>32</i>	<i>N/A</i>	<i>CCW Limit</i>
<i>BIT 6</i>	<i>64</i>	<i>CR1</i>	<i>N/A</i>
<i>BIT 7</i>	<i>128</i>	<i>CR2</i>	<i>N/A</i>

5.3 PLC Command Summary

The following is summary of the available commands and the required values:

COMMAND DESCRIPTION

- 0** **NOP** - No Operation increment to next sequence
- 1** **SWITCH ON** - LSB selects switch input MSB is branch to sequence number if switch is "OFF". If MSB is zero, function will wait for switch "ON" condition then increment to next sequence.
- 2** **SWITCH OFF** - LSB selects switch input, MSB is branch to sequence number if switch is "ON". If MSB is zero, function will wait for switch "OFF" condition then increment to next sequence.
- 3** **FORWARD LIMIT** - IF the forward limit is active branch to LSB sequence. If the limit is not active branch to sequence number specified by the MSB. If MSB is 0 the function will wait until the input becomes active. If input is active the function will branch to the sequence number specified by the LSB byte. If the LSB is 0 the function increment to the next sequence.
- 4** **REVERSE LIMIT** - IF the Reverse limit is active branch to LSB sequence. If the limit is not active branch to sequence number specified by the MSB. If MSB is 0 the function will wait until the input becomes active. If input is active the function will branch to the sequence number specified by the LSB byte. If the LSB is 0 the function increment to the next sequence.
- 5** **AUTO EXECUTE** - If the Auto execute input is active branch to LSB sequence. If the Auto Execute input is not active branch to sequence number specified by the MSB. If MSB is 0 the function will wait until the input becomes active. If input is active the function will branch to the sequence number specified by the LSB byte. If the LSB is 0 the function increment to the next sequence.
- 6** **CLEAR AUTO SEQUENCE** - Clear the auto sequence flag.

- 7 **INITIALIZE** - If the Initialize input is active branch to LSB sequence. If the Initialize input is not active branch to sequence number specified by the MSB. If MSB is 0 the function will wait until the input becomes active. If input is active the function will branch to the sequence number specified by the LSB byte. If the LSB is 0 the function increment to the next sequence.
- 8 **CLEAR INITIALIZE** - Clear initialize input.
- 9 **SET RELAY** - Set relay output specified by LSB byte (CR1 = 64 and CR2 = 128).
- 10 **CLEAR RELAY** - Clear relay output specified by LSB byte (CR1 = 64 and CR2 = 128).
- 11 **MOTOR FORWARD** - Start drive in forward direction with speed specified by the value.
- 12 **MOTOR REVERSE** - Start drive in reverse direction with speed specified by the value.
- 13 **STOP DRIVE** - Stop the drive motor.
- 14 **DELAY TIME** - Delay program execution by time specified by the value. The value specified is in 10 msecond increments. (Example: 13,50 Set delay time to 0.50 seconds)
- 15 **SET SPEED** - Value equal new drive velocity.
- 16 **SET POSITION** - Set current position encoder to value specified.
- 17 **SAVE POSTION** - Save current position encoder count to temporary position register and increment to next sequence.
- 18 **RESTORE POSITION** - Move to position saved in temporary position register and increment to next sequence.
- 19 **MOVE TO POSITION** - Move to position and increment to next sequence.
- 20 **SET MAXIMUM POSITION** - Set maximum position limit to value specified.
- 21 **LOAD POSITION TO MAX** - Set maximum position limit to current position.
- 22 **JOG POSITION FORWARD** - Add value specified to current position and move to new position.
- 23 **JOG POSITION REVERSE** - Subtract value specified from current position and move to new position.
- 24 **MOVE COMPLETE** - Has drive completed the last move command? If the MSB byte is set the function will jump to the selected sequence number if the move is not complete. If the LSB byte is zero the function will wait for the move to be completed then increment to the next sequence or branch to the sequence number set by the LSB.

- 25 SET UP LOOP COUNTER** – Load selected loop counter (0-9) with starting value. MSC byte is the value to load and the LSB byte is the selected counter. There are 10 Loop counters available.
- 26 DECREMENT LOOP COUNTER** - Decrement the selected counter (0 - 9) and set the condition code register. The LSB byte selects the loop counter to decrement.
- 27 INCREMENT LOOP COUNTER** - Increment the selected counter (0 - 9) and set the condition code register. The LSB byte selects the loop counter to Increment.
- 28 CLEAR LOOP COUNTER** - Clears the selected loop counter (0 - 9). The LSB byte selects the desired loop counter to be cleared and sets the condition code register.
- 29 JUMP TO SEQUENCE** - Jump to sequence specified by LSB byte.
- 30 JUMP SUBROUTINE** - Jump to specified sequence number subroutine. The sequence subroutine must be terminated with a return from subroutine command (32). The DMC allows nesting of up to 10 subroutines. All commands may be used in subroutines.
- 31 RETURN FROM SUBROUTINE** - Returns the sequence counter to the "jump subroutine " sequence number plus 1. If a return subroutine command is executed with out a "jump subroutine " the sequence counter will be incremented to the next sequence.
- 32 COMPARE VALUE TO POSITION** - Compares the value (MSB, LSB) to the current position and set the condition code register. The comparison is a subtraction of the current position from the command value (CURRENT POSITION - COMMAND VALUE). The current position is unchanged as a result of the command. Only the condition code register is set.
- 33 COMPARE SWITCH INPUT** - Compares the current DMC-1000 switch inputs to the binary value specified in the LSB byte (SWITCH - VALUE) and sets the condition code register.
- 34 COMPARE LOOP VALUE** - Compares the loop counter specified by the LSB byte to the value specified in the MSB byte and sets the condition code register (COUNTER - VALUE).
- 35 BRANCH NOT EQUAL** - Branch to sequence specified by LSB byte as a result of the previous parameter value not being equal to command value or not equal zero. As specified by the condition code register. If not zero increment to next sequence number.
- Function: (Parameter != Compare) then Branch to Seq #*
- 36 BRANCH IF LOWER** - Branch to sequence specified by LSB as a result of the previous parameter value being less than the command value. As specified by the condition code register. If not less than stored value increment to next sequence number.
- Function: Parameter < Compare then Branch to Seq #*

37 **BRANCH IF HIGHER** - Branch to sequence specified by LSB byte as a result of the previous parameter value being greater than the command value. As specified by the condition code register. If not greater than parameter value increment to next sequence number.

Function: Parameter > Compare then Branch to Seq #

38 **BRANCH IF EQUAL** - Branch to sequence specified by LSB byte as a result of the previous parameter value being equal to the command value (Note: result of comparison is zero). As specified by the condition code register. If not equal to parameter value increment to next sequence number.

Function: Parameter = Compare then Branch to Seq #

39 **BRANCH IF HIGHER OR EQUAL** - Branch to sequence specified by LSB byte as a result of the previous parameter value being greater than or equal to the command value. As specified by the condition code register. If not greater than command value increment to next sequence number.

Function: Parameter >= Compare then Branch to Seq #

40 **BRANCH IF LESS THEN OR EQUAL** - Branch to sequence specified by LSB byte as a result of the previous parameter value being less than or equal to the command value (Note: result of comparison is zero). As specified by the condition code register. If not equal to parameter value increment to next sequence number.

Function: Parameter <= Compare then Branch to Seq #

41 **LOAD REGISTER 0** - Load the results Register 0 with the value specified by the [MSB:LSB] bytes and set the condition code register.

42 **LOAD REGISTER 1** - Load Register 1 with the value specified by the [MSB:LSB] bytes and set the condition code register.

43 **LOAD REGISTER 2** - Load Register 2 with the value specified by the [MSB:LSB] bytes and set the condition code register.

44 **LOAD REGISTER 3** - Load Register 3 with the value specified by the [MSB:LSB] bytes and set the condition code register.

45 **LOAD REGISTER 4** - Load Register 4 with the value specified by the [MSB:LSB] bytes and set the condition code register.

46 **LOAD REGISTER 5** - Load Register 5 with the value specified by the [MSB:LSB] bytes and set the condition code register.

47 **COPY REG[MSB] TO REG[LSB]** - Load the value in Register specified by the [MSB] byte to the register specified by the [LSB] byte and set the condition code register.

48 **LOAD SPEED TO REG[LSB]** - Load the current speed parameter to the register specified by the [LSB] byte and set the condition code register.

49 **LOAD POSITION TO REG[LSB]** - Load the current drive position to the register specified by the [LSB] byte and set the condition code register.

50 **SET SPEED TO REG[LSB]** - Load the value in Register specified by the [LSB] byte to the speed parameter.

51 **MOVE POSITION IN REG[LSB]** - Load the value in Register specified by the [LSB] byte to the Move parameter and move the Drive to the new position.

52 **ADD REG[MSB] to REG[LSB]** - Add the value in REG[MSB] to the value in REG[LSB] and save the result in the results register (REG[0]) and set the condition code register. The original registers are not changed.

$$REG[0] = REG[MSB] + REG[LSB]$$

53 **SUBTRACT REG[LSB] FROM REG[MSB]** - Subtract the value in REG[LSB] from the value in REG[MSB] and save the result in the results register (REG[0]) and set the condition code register. The original registers are not changed.

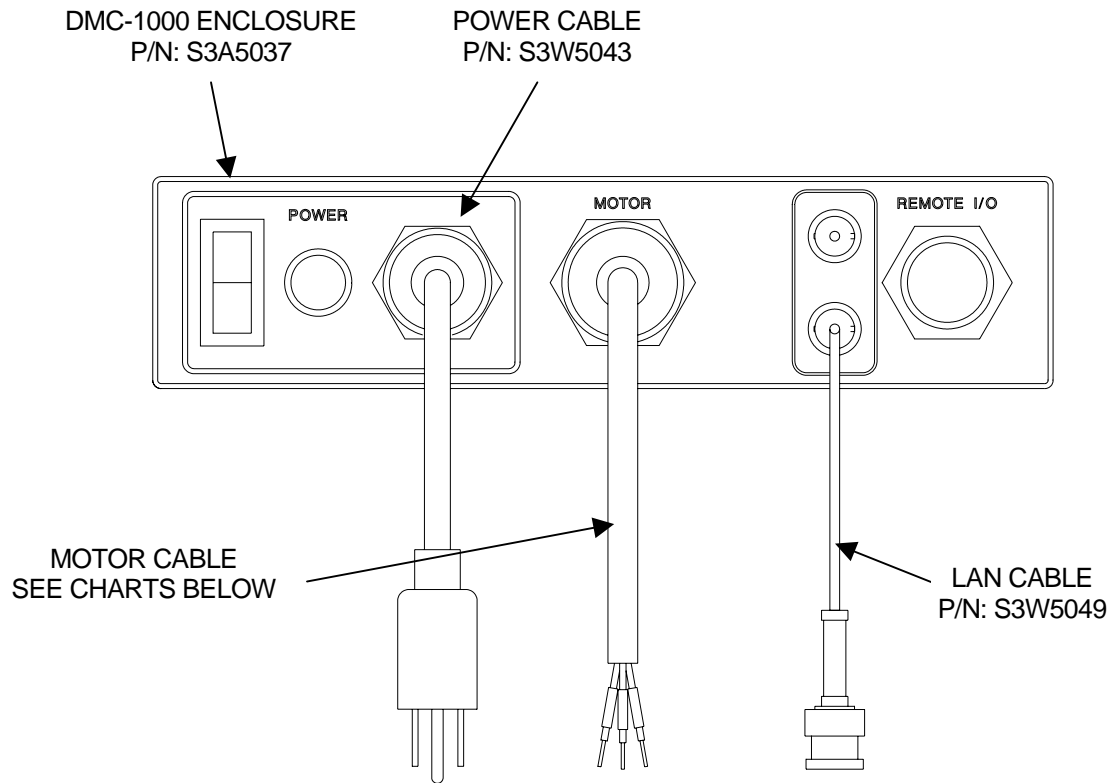
$$REG[0] = REG[MSB] - REG[LSB]$$

54 **COMPARE REG[LSB] TO POSITION** - Compare the current position to the value in REG[LSB] and set the condition code register. The specified register is not changed.

$$CCR = POSITION - REG[LSB]$$

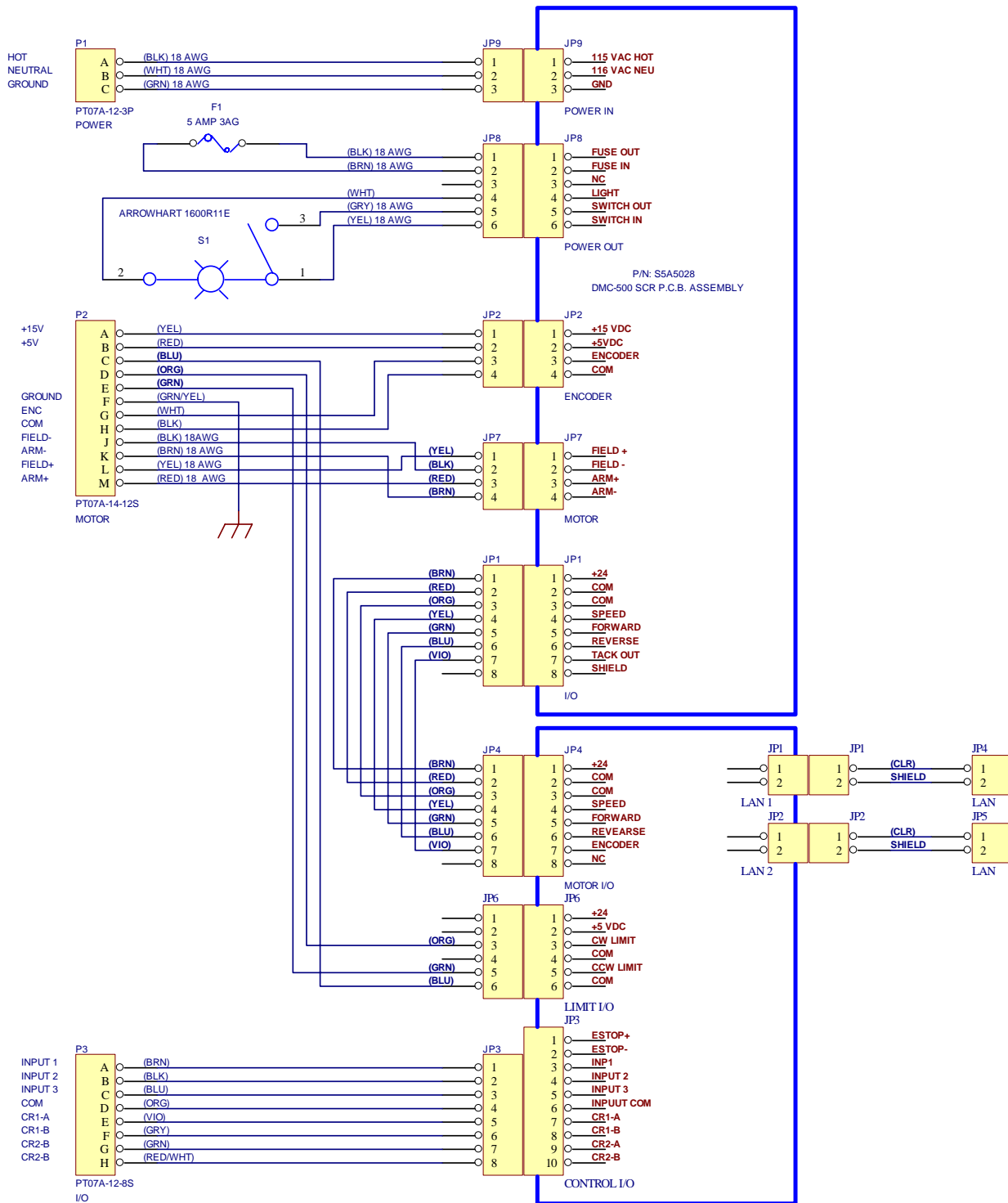
SECTION 6 – DRAWINGS AND PARTS LISTS

6.1 System Configuration

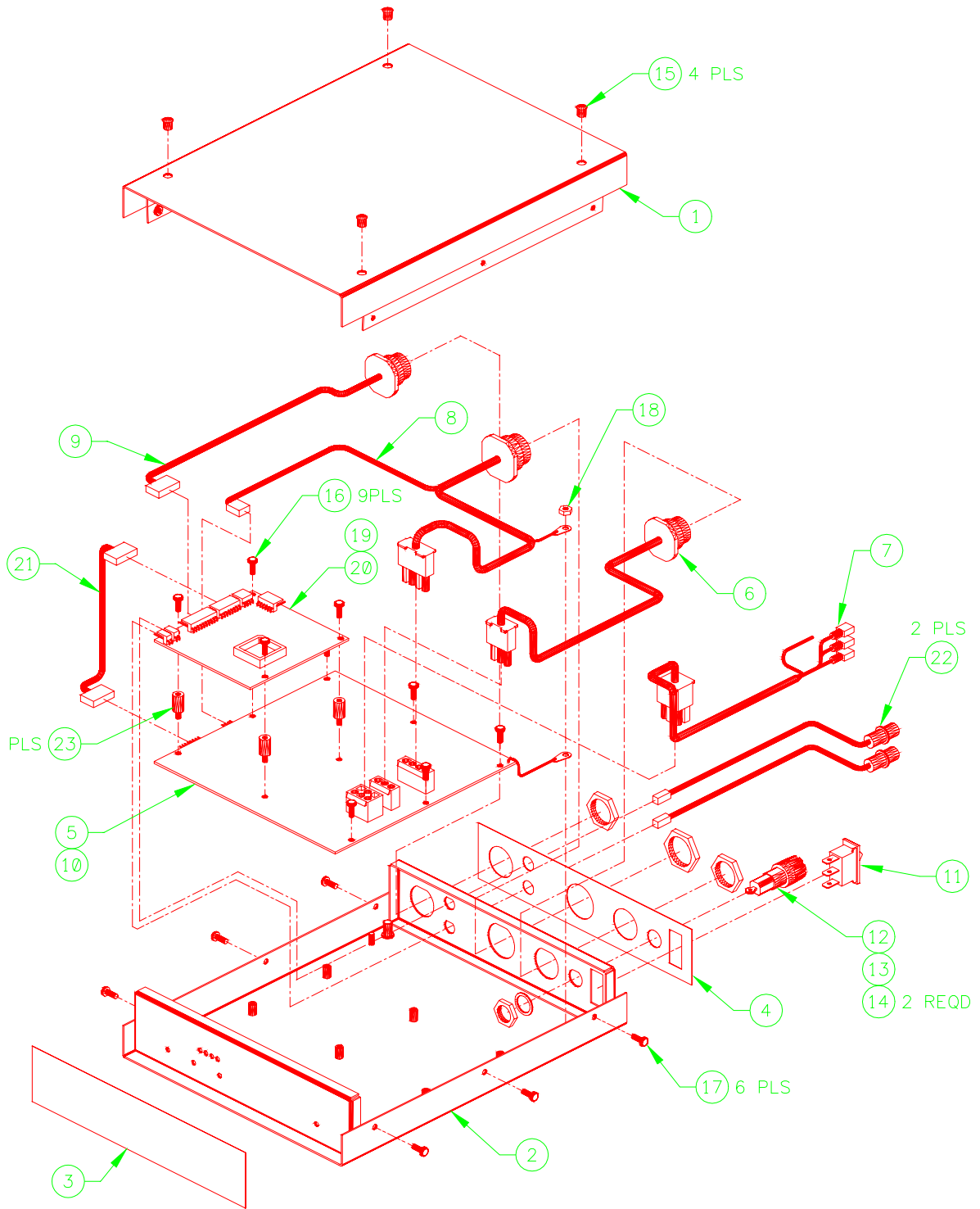


MOTOR CABLE CHART	
PART NUMBER	DESCRIPTION
S3W5059	MILLER A1D4 MOTOR CABLE
S3W5060	STANDARD MOTOR WITHOUT TACH CABLE
S3W5072	LINCOLN NA5 MOTOR CABLE
S3W5073	STANDARD MOTOR WITH TACH CABLE

6.2 System Schematic Diagram



6.3 DMC-1000 Enclosure Details



ITEM	QTY	PART NUMBER	DESCRIPTION
1	1	S3E5044	COVER
2	1	S3E5052	ENCLOSURE
3	1	S3E5059	FRONT OVERLAY
4	1	S3E5060	REAR OVERLAY
5	1	S5A5028	DMC-500 SCR PCB ASSEMBLY
6	1	S3W5054	POWER IN CABLE
7	1	S3W5055	POWER OUT CABLE
8	1	S3W5056	SCR MOTOR CABLE
9	1	S3W5057	INTERNAL I/O CABLE
10	1	S5I5013	DMC-500 SCR PROCESSOR
11	1	X3S5078	ROCKER SWITCH
12	1	X3C0003	FUSE HOLDER
13	1	X3C5021	5 AMP 3AG FUSE
14	2		¼" DIAMETER X ½" LONG HEAT SHRINK TUBING
15	4	X6Z5071	¼" HOLE DOME PLUG
16	9		#6-32 X 3/8" LONG PAN HEAD SCREW W/ INT LOCK WASHER
17	6		#6-32 X ¼" LONG SOCKET BUTTON HEAD SCREW
18	1		#6-32 HEX NUT
19	1	S5A5029	MOTOR DRIVE CPU PCB ASSEMBLY
20	1	S5I5015	DMC-1000 PROCESSOR
21	1	S3W5074	CPU I/O CABLE
22	2	S3W5075	LAN CABLE
23	4	X6S5023	#6-32 X ½" LONG M-F SPACER