Chapter

6

Modbus Communications

Contents of this Chapter...

*Sure*Servo™ Communication Parameters

The *Sure*Servo™ drives support the Modbus RTU/ASCII communications protocols as a slave device only. Drive serial port CN3 can be connected to a Modbus master using RS-232, RS-422 or RS-485 communications (port pin-outs and wiring diagrams are shown later in this chapter). This chapter lists all of the drive's parameters along with the corresponding Modbus addresses. Network masters, such as AutomationDirect PLCs, can be used to read/write drive(s) parameters.

The *Sure*Servo drive Communications Parameters listed below must be set using the *Sure*Servo Pro software or the drive keypad unless the defaults are appropriate for your application. For a detailed explanation of all *Sure*Servo Parameters, refer to Chapter 4.

Communications Parameters

Parameter	***Description***	***Range***	***Default***
P3-00	Communication Address	01 to 254	01
P3-01	Transmission Speed	00: 4800 baud 01: 9600 baud 02: 19200 baud 03: 38400 baud 04: 57600 baud 05: 115200 baud	02
P3-02	Communication Protocol	00: Modbus ASCII mode 7 data bits, no parity, 2 stop bits 01: Modbus ASCII mode 7 data bits, even parity, 1 stop bit 02: Modbus ASCII mode 7 data bits, odd parity, 1 stop bit 03: Modbus ASCII mode 8 data bits, no parity, 2 stop bits 04: Modbus ASCII mode 8 data bits, even parity, 1 stop bit 05: Modbus ASCII mode 8 data bits, odd parity, 1 stop bit 06: Modbus RTU mode 8 data bits, no parity, 2 stop bits 07: Modbus RTU mode 8 data bits, even parity, 1 stop bit 08: Modbus RTU mode 8 data bits, odd parity, 1 stop bit	08
P3-03	Transmission Fault Action	00: Display fault and continue operating 01: Display fault and RAMP to stop	00
P3-04	Communication Watchdog Time Out	0 to 20.0 seconds	00
P3-05	Communication Selection	00: RS-232 01: RS-422 02: RS-485	00
P3-06	Reserved	–	–
P3-07	Communication Response Delay Time	00 to 255ms (increments of 0.5 ms)	00

SureServo™ Parameter Memory Addresses

Parameter Memory Addresses				
Parameter	***Description***	***Hexa-decimal***	***Modbus Decimal***	***Octal***
	Group 0: Monitor Parameters			
P0-00	Software Version	0000	40001	0
P0-01	Drive Fault Code	0001	40002	1
P0-02	Drive Status (Front Panel Display)	0002	40003	2
P0-03	Analog Monitor Outputs	0003	40004	3
P0-04	Status Monitor 1	0004	40005	4
P0-05	Status Monitor 2	0005	40006	5
P0-06	Status Monitor 3	0006	40007	6
P0-07	Status Monitor 4	0007	40008	7
P0-08	Status Monitor 5	0008	40009	10
P0-09	Block Transfer Parameter 1	0009	40010	11
P0-10	Block Transfer Parameter 2	000A	40011	12
P0-11	Block Transfer Parameter 3	000B	40012	13
P0-12	Block Transfer Parameter 4	000C	40013	14
P0-13	Block Transfer Parameter 5	000D	40014	15
P0-14	Block Transfer Parameter 6	000E	40015	16
P0-15	Block Transfer Parameter 7	000F	40016	17
P0-16	Block Transfer Parameter 8	0010	40017	20
P0-17	Output Functions Status	0011	40018	21
P0-18	Servo On Time Record	0012	40019	22
	Group 1: Basic Parameters			
P1-00	External Pulse Type Input	0100	40257	400
P1-01	Control Mode and Output Direction	0101	40258	401
P1-02	Speed and Torque Limit	0102	40259	402
P1-03	Output Polarity Setting	0103	40260	403
P1-04	Analog Monitor Output Scaling 1 (ch 1)	0104	40261	404
P1-05	Analog Monitor Output Scaling 2 (ch 2)	0105	40262	405
P1-06	Analog Speed Command Low-pass Filter	0106	40263	406
P1-07	Analog Torque Command Low-pass Filter	0107	40264	407
P1-08	Position Command Low-pass Filter	0108	40265	410
P1-09	Velocity Command 1 Speed Limit 1	0109	40266	411
P1-10	Velocity Command 2 Speed Limit 2	010A	40267	412
P1-11	Velocity Command 3 Speed Limit 3	010B	40268	413
P1-12	Torque Command 1 Torque Limit 1	010C	40269	414
P1-13	Torque Command 2 Torque Limit 2	010D	40270	415
P1-14	Torque Command 3 Torque Limit 3	010E	40271	416
P1-15	Position Command 1- Revolutions	010F	40272	417
P1-16	Position Command 1- Pulse	0110	40273	420
P1-17	Position Command 2- Revolutions	0111	40274	421
P1-18	Position Command 2- Pulse	0112	40275	422
P1-19	Position Command 3- Revolutions	0113	40276	423

table continued next page

Parameter Memory Addresses (continued)				
Parameter	***Description***	***Hexa-decimal***	***Modbus Decimal***	***Octal***
P1-20	Position Command 3- Pulse	0114	40277	424
P1-21	Position Command 4- Revolutions	0115	40278	425
P1-22	Position Command 4- Pulse	0116	40279	426
P1-23	Position Command 5- Revolutions	0117	40280	427
P1-24	Position Command 5- Pulse	0118	40281	430
P1-25	Position Command 6- Revolutions	0119	40282	431
P1-26	Position Command 6- Pulse	011A	40283	432
P1-27	Position Command 7- Revolutions	011B	40284	433
P1-28	Position Command 7- Pulse	011C	40285	434
P1-29	Position Command 8- Revolutions	011D	40286	435
P1-30	Position Command 8- Pulse	011E	40287	436
P1-31	Motor Code	011F	40288	437
P1-32	Motor Stop Code	0120	40289	440
P1-33	Position Control Mode	0121	40290	441
P1-34	Acceleration Time	0122	40291	442
P1-35	Deceleration Time	0123	40292	443
P1-36	Acceleration/Deceleration S-curve	0124	40293	444
P1-37	Inertia Mismatch Ratio	0125	40294	445
P1-38	Zero Speed Output Threshold	0126	40295	446
P1-39	Target Speed Output Threshold	0127	40296	447
P1-40	Analog Full Scale Velocity Command/Limit	0128	40297	450
P1-41	Analog Full Scale Torque Command/Limit	0129	40298	451
P1-42	On Delay Time of Electromagnetic Brake	012A	40299	452
P1-43	Off Delay Time of Electromagnetic Brake	012B	40300	453
P1-44	Electronic Gear Numerator 1	012C	40301	454
P1-45	Electronic Gear Denominator	012D	40302	455
P1-46	Encoder Output Scaling Factor	012E	40303	456
P1-47	Homing Mode	012F	40304	457
P1-48	Homing Speed 1 - Fast Search Speed	0130	40305	460
P1-49	Homing Speed 2 - Creep Speed	0131	40306	461
P1-50	Home Position Offset (revolutions)	0132	40307	462
P1-51	Home Position Offset (counts)	0133	40308	463
P1-52	Regenerative Resistor Value	0134	40309	464
P1-53	Regenerative Resistor Capacity	0135	40310	465
P1-54	In Position Window	0136	40311	466
P1-55	Maximum Speed Limit	0137	40312	467
P1-56	Overload Output Warning Threshold	0138	40313	470

table continued next page

Parameter Memory Addresses (continued)				
Parameter	**Description**	**Hexa-decimal**	**Modbus Decimal**	**Octal**
	Group 2: Extended Parameters			
P2-00	Position Loop Proportional Gain (KPP)	0200	40513	1000
P2-01	Position Loop Gain Boost	0201	40514	1001
P2-02	Position Feed Forward Gain (KFF)	0202	40515	1002
P2-03	Smooth Constant of Position Feed Forward Gain	0203	40516	1003
P2-04	Velocity Loop Proportional Gain (KVP)	0204	40517	1004
P2-05	Velocity Loop Gain Boost	0205	40518	1005
P2-06	Velocity Loop Integral Compensation	0206	40519	1006
P2-07	Velocity Feed Forward Gain	0207	40520	1007
P2-08	Factory Defaults and Security	0208	40521	1010
P2-09	Debounce Filter	0209	40522	1011
P2-10	Digital Input Terminal 1 (DI1)	020A	40523	1012
P2-11	Digital Input Terminal 2 (DI2)	020B	40524	1013
P2-12	Digital Input Terminal 3 (DI3)	020C	40525	1014
P2-13	Digital Input Terminal 4 (DI4)	020D	40526	1015
P2-14	Digital Input Terminal 5 (DI5)	020E	40527	1016
P2-15	Digital Input Terminal 6 (DI6)	020F	40528	1017
P2-16	Digital Input Terminal 7 (DI7)	0210	40529	1020
P2-17	Digital Input Terminal 8 (DI8)	0211	40530	1021
P2-18	Digital Output Terminal 1 (DO1)	0212	40531	1022
P2-19	Digital Output Terminal 2 (DO2)	0213	40532	1023
P2-20	Digital Output Terminal 3 (DO3)	0214	40533	1024
P2-21	Digital Output Terminal 4 (DO4)	0215	40534	1025
P2-22	Digital Output Terminal 5 (DO5)	0216	40535	1026
P2-23	Notch Filter (resonance suppression)	0217	40536	1027
P2-24	Notch Filter Attenuation (resonance suppress.)	0218	40537	1030
P2-25	Low-pass Filter (resonance suppression)	0219	40538	1031
P2-26	External Anti-Interference Gain	021A	40539	1032
P2-27	Gain Boost Control	021B	40540	1033
P2-28	Gain Boost Switching Time	021C	40541	1034
P2-29	Gain Boost Switching Condition	021D	40542	1035
P2-30	Auxiliary Function	021E	40543	1036
P2-31	Auto and Easy Mode Response Level	021F	40544	1037
P2-32	Tuning Mode	0220	40545	1040
P2-34	Overspeed Fault Threshold	0222	40547	1042
P2-35	Position Deviation Fault Window	0223	40548	1043
P2-36	Position 1 Velocity	0224	40549	1044
P2-37	Position 2 Velocity	0225	40550	1045
P2-38	Position 3 Velocity	0226	40551	1046
P2-39	Position 4 Velocity	0227	40552	1047
P2-40	Position 5 Velocity	0228	40553	1050
P2-41	Position 6 Velocity	0229	40554	1051
P2-42	Position 7 Velocity	022A	40555	1052
P2-43	Position 8 Velocity	022B	40556	1053
P2-44	Digital Output Mode	022C	40557	1054
P2-45	Index Mode Output Signal Delay Time	022D	40558	1055
P2-46	Index Mode - Stations	022E	40559	1056
P2-47	Position Deviation Clear Delay Time	022F	40560	1057
P2-48	Backlash Compensation (index mode)	0230	40561	1060

table continued next page

Parameter Memory Addresses (continued)				
Parameter	***Description***	***Hexa-decimal***	***Modbus Decimal***	***Octal***
P2-49	Jitter Suppression	0231	40562	1061
P2-50	Clear Position Mode	0232	40563	1062
P2-51	Servo On Command	0233	40564	1063
P2-52	Dwell Time 1 (auto index mode)	0234	40565	1064
P2-53	Dwell Time 2 (auto index mode)	0235	40566	1065
P2-54	Dwell Time 3 (auto index mode)	0236	40567	1066
P2-55	Dwell Time 4 (auto index mode)	0237	40568	1067
P2-56	Dwell Time 5 (auto index mode)	0238	40569	1070
P2-57	Dwell Time 6 (auto index mode)	0239	40570	1071
P2-58	Dwell Time 7 (auto index mode)	023A	40571	1072
P2-59	Dwell Time 8 (auto index mode)	023B	40572	1073
P2-60	Electronic Gear Numerator 2	023C	40573	1074
P2-61	Electronic Gear Numerator 3	023D	40574	1075
P2-62	Electronic Gear Numerator 4	023E	40575	1076
P2-63	Velocity and Position Deviation Scaling Factor	023F	40576	1077
P2-64	Advanced Torque Limit Mode	0240	40577	1100
P2-65	Special Input Functions	0241	40578	1101
Group 3: Communication Parameters				
P3-00	Communication Address	0300	40769	1400
P3-01	Transmission Speed	0301	40770	1401
P3-02	Communication Protocol	0302	40771	1402
P3-03	Communication Fault Action	0303	40772	1403
P3-04	Communication Watchdog Time Out	0304	40773	1404
P3-05	Communication Selection	0305	40774	1405
P3-07	Communication Response Delay Time	0307	40776	1407
P3-08	Digital Input Software Control Mask	0308	40777	1410
Group 4: Diagnostic Parameters				
P4-00	Fault Record (N) (most recent)	0400	41025	2000
P4-01	Fault Record (N-1)	0401	41026	2001
P4-02	Fault Record (N-2)	0402	41027	2002
P4-03	Fault Record (N-3)	0403	41028	2003
P4-04	Fault Record (N-4)	0404	41029	2004
P4-05	Jog Function	0405	41030	2005
P4-06	Force Outputs Command	0406	41031	2006
P4-07	Input Status	0407	41032	2007
P4-09	Output Status	0409	41034	2011
P4-20	Analog Monitor 1 Offset (ch 1)	0414	41045	2024
P4-21	Analog Monitor 2 Offset (ch 2)	0415	41046	2025
P4-22	Analog Velocity Input Offset	0416	41047	2026
P4-23	Analog Torque Input Offset	0417	41048	2027

Connecting to *Direct*Logic PLCs

The following steps explain how to connect and communicate with the *Sure*Servo drives using *Direct*LOGIC PLCs.

Step 1: Modbus RTU Master PLCs

The *Sure*Servo™ servo drives will communicate with the following ***Direct***LOGIC CPUs using the Modbus RTU protocol.

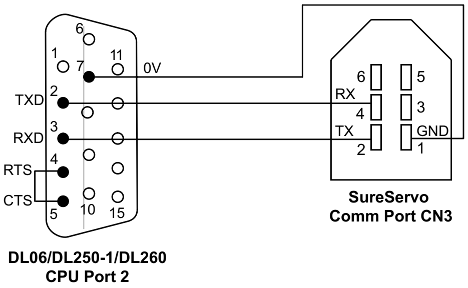
Modbus RTU Master Support	
MRX/MWX Instructions	DL06 or DL-260 CPU port 2
RX/WX Instructions	DL05, DL06, DL250-1 or DL260 CPU port 2

Step 2: Make the Connections

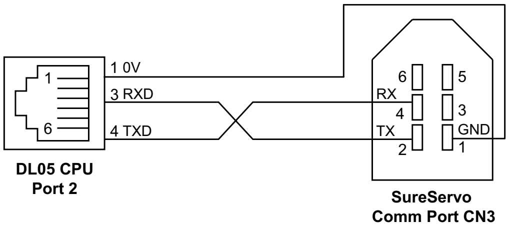
There are several means of communicating serially from a ***Direct***logic PLC.

CPU Connections	
RS-232	DL05/DL06/DL250-1/DL260 port 2
RS-485	DL06/DL260 port 2
RS-422	DL06/DL250-1/DL260 port 2

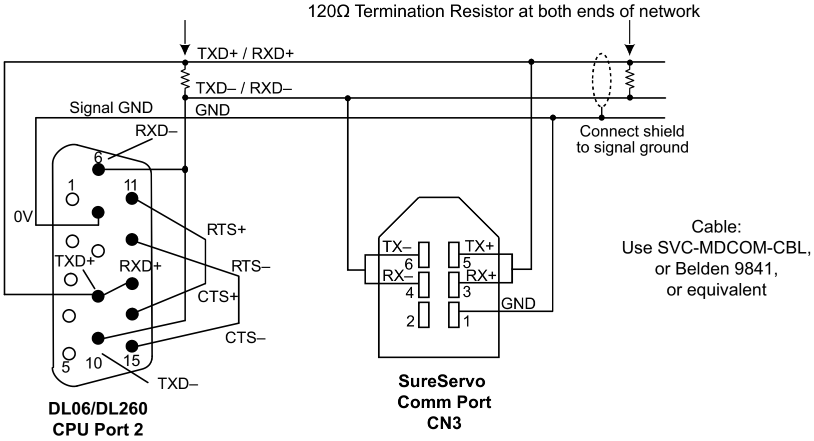
DL06/DL250-1/DL260: RS-232 Connection Wiring



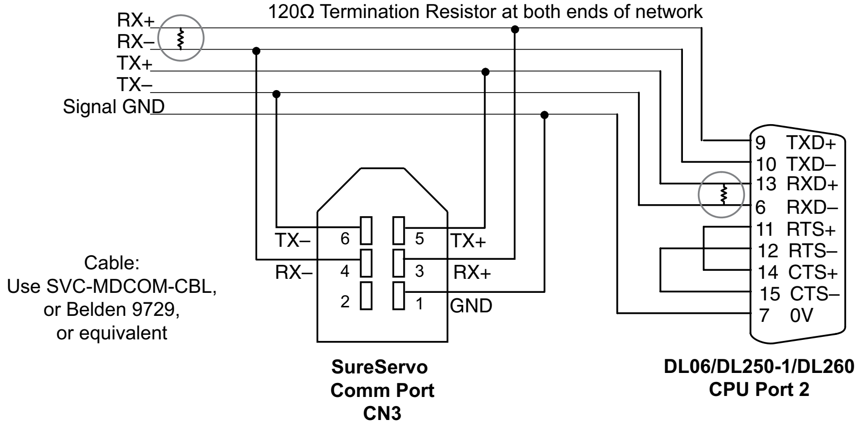
DL05: RS-232 Connection Wiring



DL06/DL260: RS-485 Connection Wiring



DL06/DL250-1/DL260: RS-422 Connection Wiring



Termination Resistors are required at both ends of RS-422/485 networks. It is necessary to select resistors that match the impedance rating of the cable (between 100 and 500 Ohms.)

SureServo drives have a provision for shutting down control or power to the drive in the event of a communications timeout. This is set up using drive parameters P3-03 and P3-04 along with a digital output configured for servo fault alarm.

Step 3: Confirm/Set Servo Communication Parameters

Most drive parameters can be written to or updated from a master controller using Modbus communications. However, the drive's operational "run" commands (i.e Servo On, Command Trigger, RESET, etc) can only be executed by controlling the drive's physical digital inputs.

The following *Sure*Servo™ communications parameters must match the ***Direct***LOGIC CPU port settings in order to establish communications. Refer to the servo Communication parameters (P3-**) for available settings.

P3-00: Communication address (default 1) - PLC read/write instructions use comm address to target a specific drive

P3-01: Communication baud rate (default 19200 bps)

P3-02: Communication protocol (default Modbus RTU mode <8 data bits, odd parity, 1 stop bit>

P3-05: Communication Selection (default RS-232)

Other related Parameters to note:

P2-30: Aux Function - setting this parameter to (5) will disable "parameter write to EEPROM" each time communications is attempted with the drive (default 0). This parameter setting is not retained when power is disconnected from the drive.

The previous list of parameter settings is the minimum required to establish communications with a DirectLOGIC PLC. There are several other parameters that must be set through the drive keypad to configure the drive up for your application.

Step 4: Configure the DirectLOGIC CPU Port 2

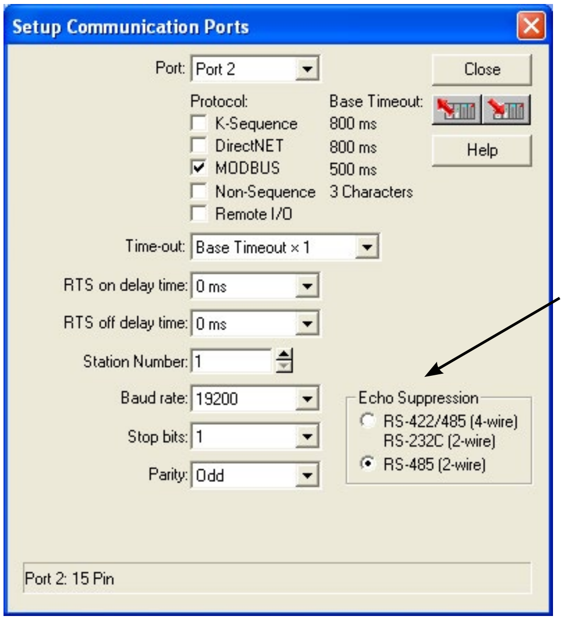
The ***Direct***LOGIC CPUs must be configured as a Modbus RTU master PLC to communicate with the *Sure*Servo drives. This includes setting up the PLC communication port parameters and creating ladder logic programming code that uses read/write instructions to communicate with the drive(s).

The set up for all of the ***Direct***LOGIC CPUs is very similar. Refer to the appropriate CPU User Manual for the specifics on your ***Direct***LOGIC CPU.

*Direct*LOGIC Modbus RTU Master Port Configuration for DL06/DL260

The following configuration example is specific to the DL06/DL260 CPU. Refer to the appropriate CPU User Manual for the specifics on your ***Direct***LOGIC CPU.

- *In **Direct**SOFT, select the PLC menu, then Setup, then "Secondary Comm Port"*
- *From the Port list box, select "Port 2"*
- *For the Protocol, select "Modbus"*
- *In the Timeout list box, select "800 ms"*
- *Response Delay Time should be "0 ms"*
- *The Station Number should be set to "1" to allow the CPU to function as network master*
- *The Baud Rate should be set at "19200"*
- *In the Stop Bits list box, select "1"*
- *In the Parity list box, select "Odd"*
- *In the Echo Suppression box, select the wiring method used in the application*



Port: Port 2

Protocol: Base Timeout:

☐ K-Sequence 800 ms
☐ DirectNET 800 ms
☑ MODBUS 500 ms
☐ Non-Sequence 3 Characters
☐ Remote I/O

Time-out: Base Timeout x 1

RTS on delay time: 0 ms

RTS off delay time: 0 ms

Station Number: 1

Baud rate: 19200

Stop bits: 1

Parity: Odd

Echo Suppression
○ RS-422/485 (4-wire)
RS-232C (2-wire)
● RS-485 (2-wire)

Port 2: 15 Pin

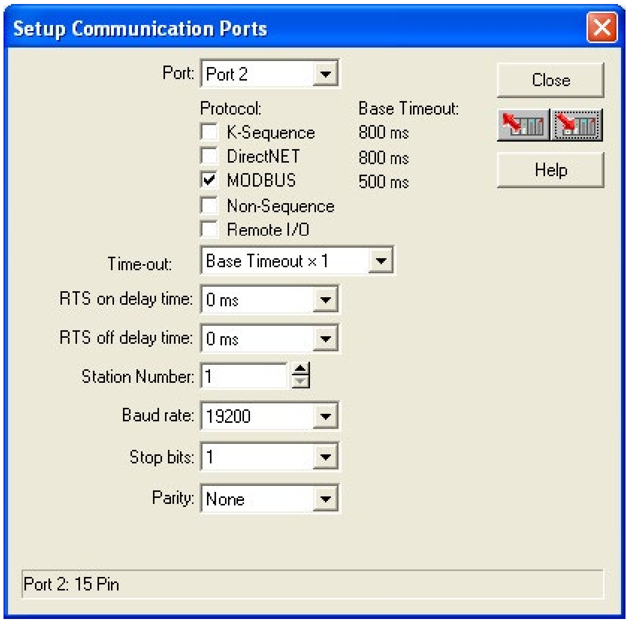
Select the appropriate button based on the comm wiring

DirectLOGIC Modbus RTU Master Port Configuration for DL05/DL250-1

The following configuration example is specific to the DL05 or DL250-1 CPU. Refer to the appropriate CPU User Manual for the specifics on your ***Direct***LOGIC CPU.

- *In* ***Direct****SOFT, select the PLC menu, then Setup, then "Secondary Comm Port"*
- *From the Port list box, select "Port 2"*
- *For the Protocol, select "Modbus"*
- *In the Timeout list box, select "800 ms"*
- *Response Delay Time should be "0 ms"*
- *The Station Number should be set to "1" to allow the CPU to function as network master*
- *The Baud Rate should be set at "19200"*
- *In the Stop Bits list box, select "1"*
- *In the Parity list box, select "Odd"*

The DL05/DL250-1 network instructions used in Master mode will access only slaves 1 to 90. Each slave must have a unique number.



Setup Communication Ports

Port: Port 2

Protocol:	Base Timeout:
☐ K-Sequence	800 ms
☐ DirectNET	800 ms
☑ MODBUS	500 ms
☐ Non-Sequence	
☐ Remote I/O	

Time-out: Base Timeout x 1

RTS on delay time: 0 ms

RTS off delay time: 0 ms

Station Number: 1

Baud rate: 19200

Stop bits: 1

Parity: None

Port 2: 15 Pin

*Sure*Servo™ / *Direct*LOGIC PLC Control Example

*Sure*Servo™ Block Transfer Function

A group of Status Monitor Registers (P0-04 to P0-08) and a group of Block Data Registers (P0-09 to P0-16) are available in the *Sure*Servo drive. These continuous blocks of registers can be used to "group" miscellaneous drive parameters together allowing you to read/write the desired parameters in one block instead of having to use a Read/Write command for each parameter.

*Sure*Servo™ Drive Parameter Settings Example – Position Mode

The parameters listed below must be entered through the drive keypad or *Sure*Servo™ Pro software in order for the provided ladder logic example to function properly. (Parameters marked with * must be entered from the drive keypad only.) Prior to configuring a new *Sure*Servo drive or re-configuring an existing drive for a new application, it is recommended to set P2-08 = 10, then cycle drive power. This will reset drive parameters to factory defaults.

P1-01 = 101: sets drive to position mode with internal control
P1-33 = 1: sets drive to incremental mode
P1-34 = 500: sets the accel time to 500ms
P1-35 = 500: sets the decel time to 500ms
P1-36 = 1000: >1 to allow the accel and decel to operate

Read transfer block from drive

P0-04 = 1: assigns motor feedback rotation to Status Monitor 1
P0-05 = 0: sets the motor feedback pulse to Status Monitor 2
P0-06 = 6: assigns motor rpm to Status Monitor 3
P0-07 = 11: assigns current % load to Status Monitor 4
P0-08 = 12: assigns peak % load to Status Monitor 5
* P0-09 = 409: assigns the digital output word to Block Transfer 1
* P0-10 = 407: assigns the digital input word to Block Transfer 2

Write transfer block to drive

* P0-11 = 21E: assigns Aux Function EEPROM write control to Block Transfer 3
* P0-12 = 10F: assigns the 1st position command revolution word to Block Trans 4
* P0-13 = 110: assigns the 1st position command pulse word to Block Transfer 5
* P0-14 = 224: assigns the 1st position velocity reference to Block Transfer 6
P2-10 = 101: assigns digital input 1 to Servo On bit
P2-11 = 108: assigns digital input 2 to Command Trigger bit
P2-12 = 104: assigns digital input 3 Pulse Clear
P2-13 = 111: assigns digital input 4 Position Zero
P2-14 = 102: assigns digital input 5 to Reset bit
P2-15 = 22: assigns digital input 6 to CWL limit (NC)
P2-16 = 23: assigns digital input 7 to CCWL limit (NC)
P2-17 = 21: assigns digital input 8 to External Fault (NC)
P2-18 = 101: assigns digital output 1 to Servo Ready
P2-19 = 103: assigns digital output 2 to Low Speed
P2-20 = 109: assigns digital output 3 to Home Search
P2-21 = 105: assigns digital output 4 to In Position
P2-22 = 7: assigns digital output 5 to Servo Fault (NC)

*** These parameters must be entered using the drive keypad.**

The following list provides the DirectLOGIC PLC V-memory locations and control bits along with the associated SureServo parameters used in the following ladder logic drive control example.

Parameters Read from drive (RX) and Placed in PLC V-memory

V3000 - P0-00: Firmware Version
V3001 - P0-01: Drive fault
V3002 - P0-02: Drive Status
V3003 - P0-03: Analog Monitor Output
V3004 - P0-04: Motor Feedback Rotation
V3005 - P0-05: Motor Feedback Pulse
V3006 - P0-06: Motor RPM
V3007 - P0-07: Current Load (% of rated torque)
V3010 - P0-08: Peak Load (% of rated torque since powerup)
V3011 - P0-09: Digital Output Word
V3012 - P0-10: Digital Input Word
V3013 - P0-11: Read drive EEPROM control value

Parameters/Values Written to drive (WX) from PLC V-memory

V2000 - P0-11: Drive write to EEPROM control
V2001 - P0-12: Position Command Revolutions
V2002 - P0-13: Position Command pulse
V2003 - P0-14: Velocity Reference (rpm)
V2013 - User memory location to compare velocity reference and update

Drive's digital outputs mapped from V3011 to VC120

C120 - P2-18: Digital output 1 - Servo Ready
C121 - P2-19: Digital output 2 - Low Speed
C122 - P2-20: Digital output 3 - Home Search
C123 - P2-21: Digital output 4 - In position
C124 - P2-22: Digital output 5 - Servo Fault (normally closed)

Drive's digital input terminals connected to PLC discrete outputs

Digital Input 1 - SERVO ENABLE
Digital Input 2 - CMD TRIGGER
Digital Input 3 - Pulse Clear
Digital Input 4 - Position Zero
Digital Input 5 - RESET
Digital Input 6 - CWL Limit (normally closed)
Digital Input 7 - CCWL Limit (normally closed)
Digital Input 8 - External Fault (normally closed)

*Direct*LOGIC Ladder Logic Programming Example

The setup for all of the *Direct*LOGIC CPUs is very similar. **Refer to the appropriate CPU User Manual for the specifics on your particular *Direct*LOGIC CPU model**.

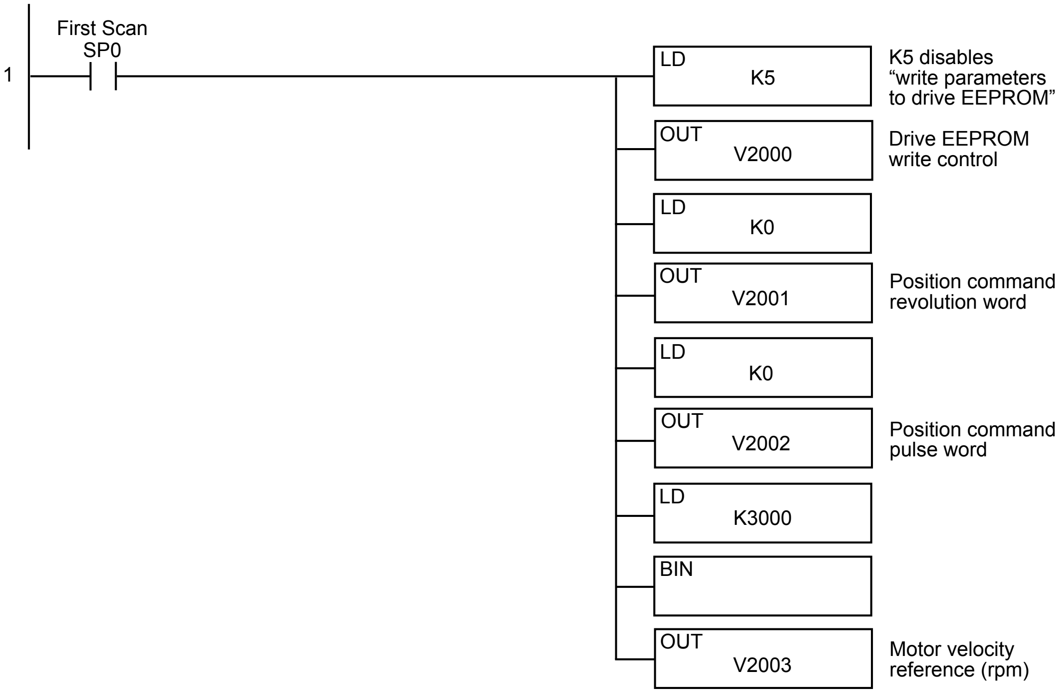
The following ladder program shows an example of how to control the *Sure*Servo drive (configured for Position Mode) using communications instructions via the Modbus RTU protocol. The drive should be set up and tested for communications before it is connected to a load.

WARNING: A DRIVE SHOULD NEVER BE CONNECTED TO A LOAD UNTIL ANY APPLICABLE COMMUNICATION PROGRAMS HAVE BEEN PROVEN.

This program is for example purposes only and not intended for a specific application. The drive parameters listed on the previous pages are required for the following example program to function properly.

DirectLOGIC Ladder Logic Programming Example

Rung 1 initializes the drive on first scan. The motor pulse and revolutions registers are set to zero and the motor velocity reference is set to 3000rpm.

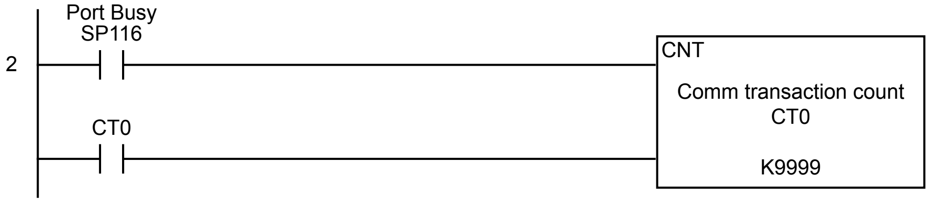

First Scan
SP0
1
LD K5
K5 disables "write parameters to drive EEPROM"
OUT V2000
Drive EEPROM write control
LD K0
OUT V2001
Position command revolution word
LD K0
OUT V2002
Position command pulse word
LD K3000
BIN
OUT V2003
Motor velocity reference (rpm)

(example program continued next page)

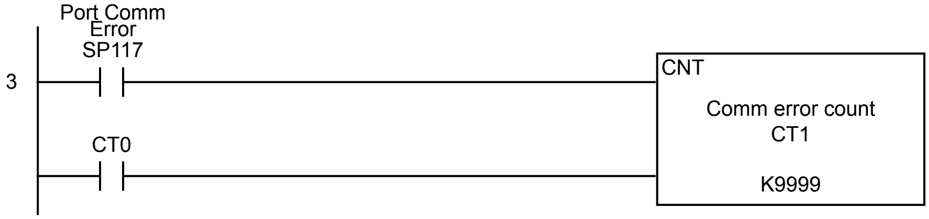
DirectLOGIC Ladder Logic Programming Example (continued)

In many drive applications, electromagnetic interference can at times cause frequent, short duration, communication errors. Unless the application environment is perfect, an occasional communication error will occur. In order to distinguish between these non-fatal transients and a genuine communication failure, you may want to use the instructions as shown in Rungs 2 and 3.

Rung 2 monitors the number of times that the PLC attempts to communicate with the drive. When the PLC's communication attempts are successful, SP116 (port busy) will count up and SP117 (comm error) will not count. Once the count reaches 9999, the counter will reset and resume counting.



Rung 3 monitors the number of times the PLC fails in communicating with the drive.



Alternative resets/control bits can be used in your application program.

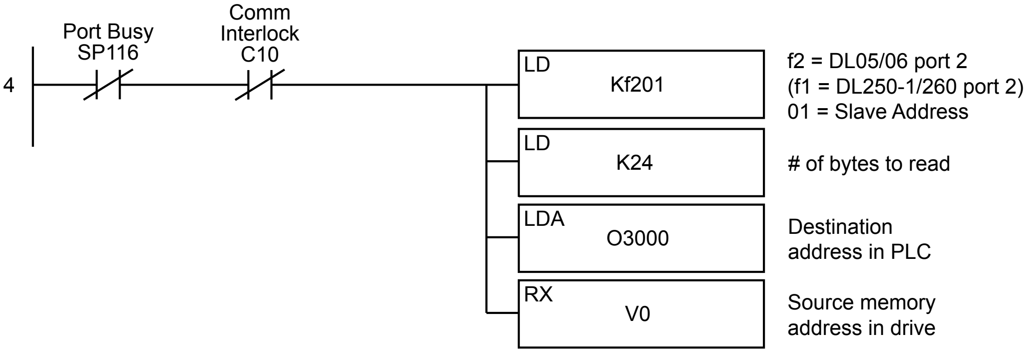
(example program continued next page)

DirectLOGIC Ladder Logic Programming Example (continued)

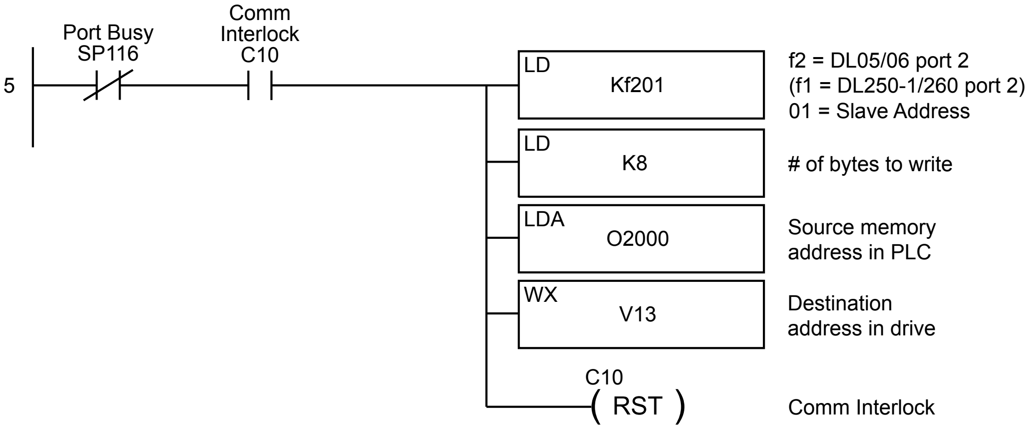
The Read(RX) and Write(WX) commands are supported in the DL05/06/250-1/260 DirectLOGIC CPUs. These instructions use octal addressing only, so the octal equivalent of the Parameter's Modbus addresses must be used.

Rungs 4 & 5 for Read and Write using RX and WX commands

Rung 4 reads the first 12 Monitor Parameters (P0-00 to P0-11) in the drive and places the values in V3000 - V3013 in the PLC. (Octal V0 - V13 equals Modbus 40001 - 40012).



Rung 5 writes 4 words (V2000 - V2003) from the PLC to drive Block Read/Write registers P0-11 to P0-14 (Octal V13 - V16 equals Modbus 40012 - 40015).



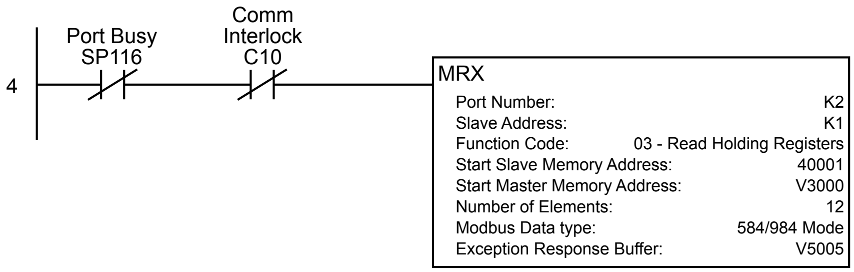
(example program continued next page)

DirectLOGIC Ladder Logic Programming Example (continued)

The DL06/260 CPUs support the Modbus Read (MRX) and Modbus Write (MWX) instructions. These instructions allow you to enter Modbus Slave Memory Addresses (no need to use octal addressing conversions to communicate with the drive).

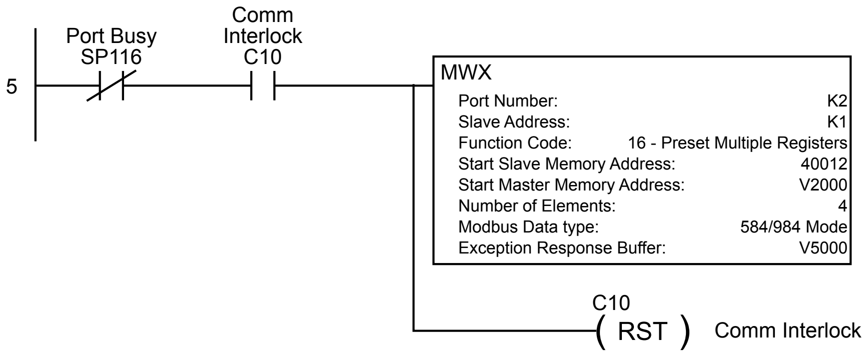
Alternate Rungs 4 & 5 for Read and Write using MRX and MWX commands with DL06/DL260 PLCs

Rung 4 reads the first 12 (P0-00 to P0-11) Monitor Parameters from the drive and places the values in V3000 - V3013 in the PLC.



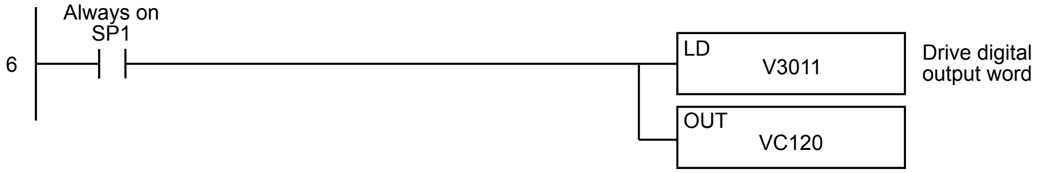
Port Number:	K2
Slave Address:	K1
Function Code:	03 - Read Holding Registers
Start Slave Memory Address:	40001
Start Master Memory Address:	V3000
Number of Elements:	12
Modbus Data type:	584/984 Mode
Exception Response Buffer:	V5005

Rung 5 writes 4 words (V2000 - V2003) from the PLC to drive Block Transfer Registers P0-11 - P0-14 (Modbus 40012 - 40015).

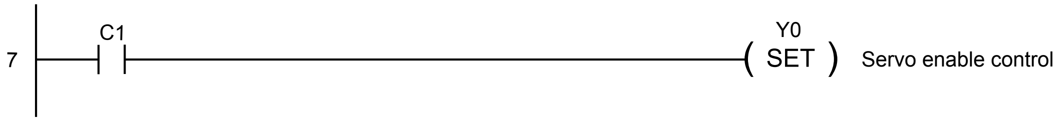


Port Number:	K2
Slave Address:	K1
Function Code:	16 - Preset Multiple Registers
Start Slave Memory Address:	40012
Start Master Memory Address:	V2000
Number of Elements:	4
Modbus Data type:	584/984 Mode
Exception Response Buffer:	V5000

Rung 6 maps the drive's digital output word that was read using the RX or MRX instruction from V3011 to C120 - C124 for bit level use.



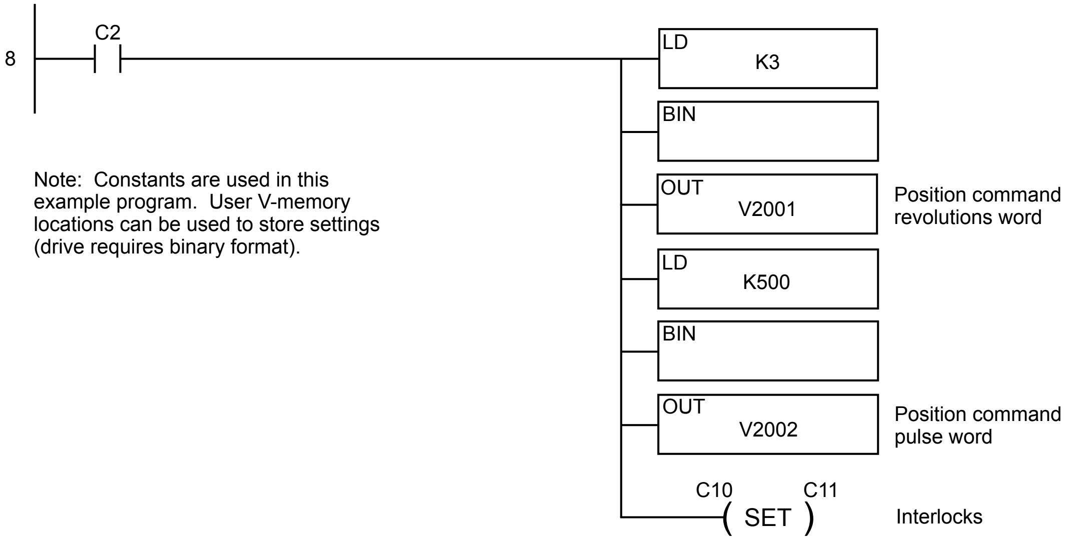
Rung 7 enables the drive (digital input 1 = Servo Enable) when C1 is turned on. Y0 is connected to drive digital input 1.



(example program continued next page)

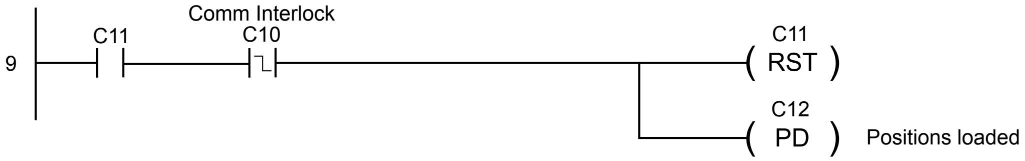
DirectLOGIC Ladder Logic Programming Example (continued)

Rung 8 loads the position (revolutions and pulse) counts to the drive when C2 is turned on. The registers are written by the WX or MWX instruction.

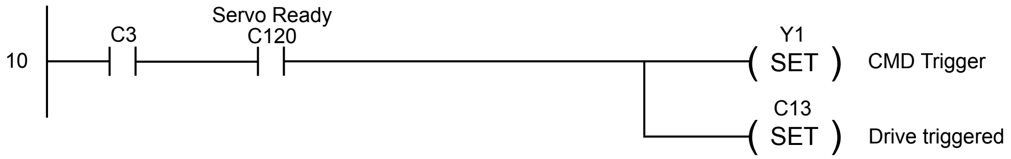

8
C2
LD
K3
BIN
OUT
V2001
Position command revolutions word
LD
K500
BIN
OUT
V2002
Position command pulse word
C10
SET
C11
Interlocks

Note: Constants are used in this example program. User V-memory locations can be used to store settings (drive requires binary format).

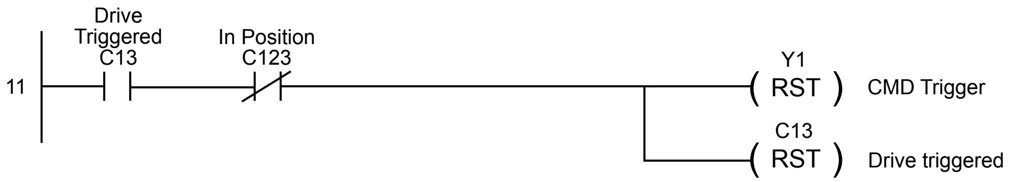
Rung 9: C12 is triggered once the Position is loaded into the drive.



Rung 10 sets the drive’s Command Trigger input to begin the motor position movement and sets C13, the drive triggered bit. Y1 is connected to drive digital input 2.



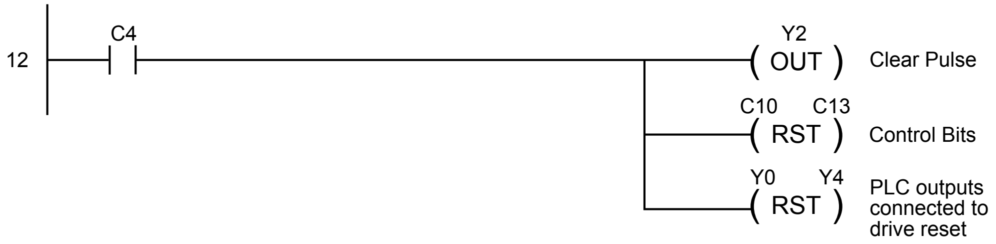
Rung 11: If the drive has been triggered and is not in position (motor is moving), the drive input CMD trigger and drive triggered flag are reset.



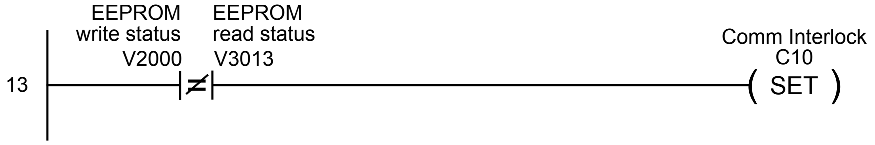
(example program continued next page)

DirectLOGIC Ladder Logic Programming Example (continued)

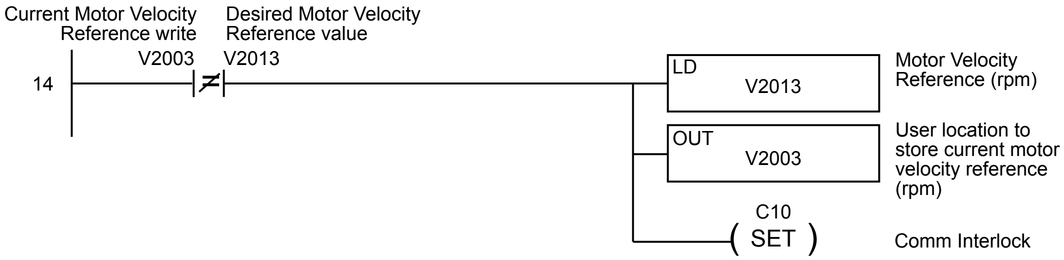
Rung 12: If C4 is turned on, drive faults and the ladder logic is reset. Y2 is connected to drive input 3. Y4 is connected to drive input 5.



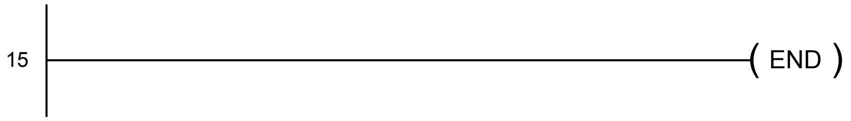
Rung 13: If the EEPROM write control register (V2000) is not equal to the value read (RX or MRX) and stored in V3013, C10 will be set to enable the WX or MWX command (rung 5). This will update the drive with the value in V2000.
For example, drive parameter P2-30 (write to EEPROM control) is not retentive during drive power cycle, so the read value stored in V3013 will be 0 (zero) and the value in V2000 may be (5). This will enable the rung 13 and cause rung 5 to execute the write to drive transfer block.



Rung 14: If the motor velocity reference register (V2003) is not equal to the previous velocity value stored in V2013 (user V-memory location), the WX command (rung 4) will execute and write the new velocity reference to the drive and will map the current value (V2003) to user V-memory location V2013.


Current Motor Velocity Reference write V2003
Desired Motor Velocity Reference value V2013
14
≠
LD V2013
Motor Velocity Reference (rpm)
OUT V2003
User location to store current motor velocity reference (rpm)
C10
SET
Comm Interlock

Rung 15: All ladder logic programs must be terminated with an (END) command.



*Direct*LOGIC Ladder Programming Example – Multiple Drives

The set up for all of the ***Direct***Logic CPUs is very similar. Refer to the appropriate CPU User Manual for the specifics on your ***Direct***Logic CPU.

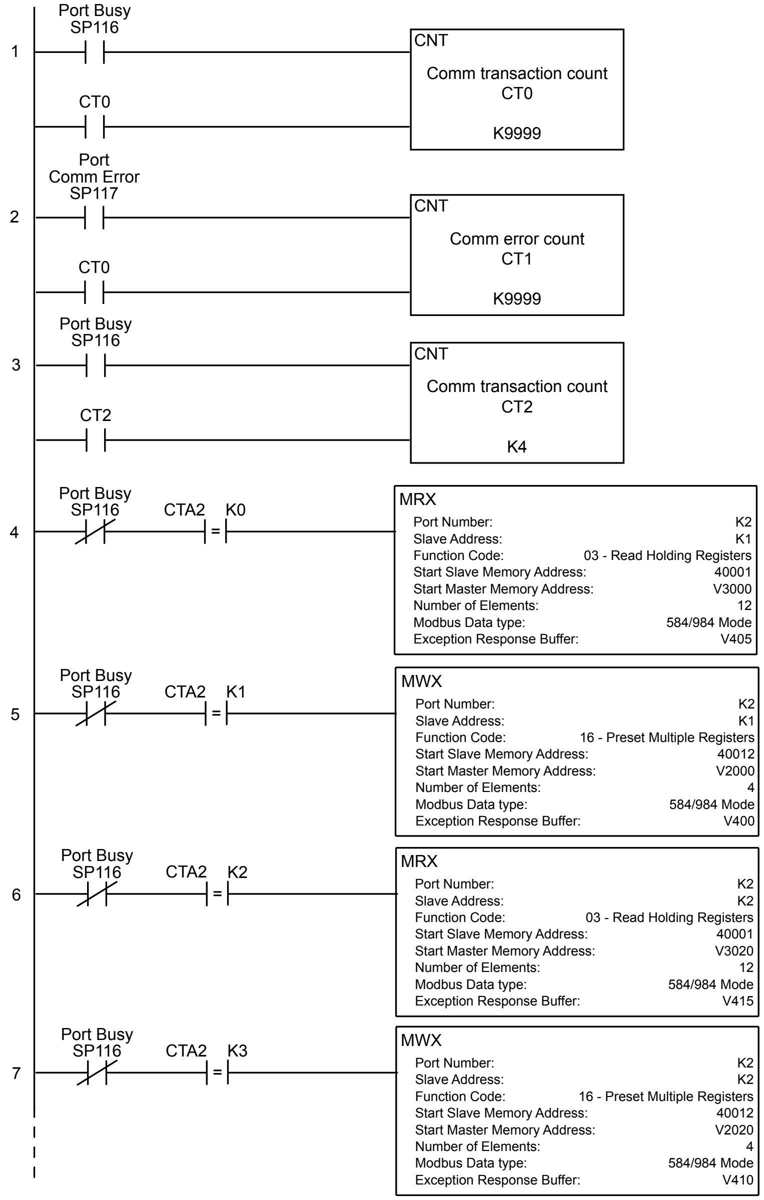
The following ladder program shows an example of a DL06 or DL260 CPU port 2 controlling two *Sure*Servo™ drives using MRX/MWX instructions. The drive must be set up and tested for communications before it is connected to a load. See the previous ladder example for rung instruction explanations.

WARNING: A DRIVE SHOULD NEVER BE CONNECTED TO A LOAD UNTIL ANY APPLICABLE COMMUNICATION PROGRAMS HAVE BEEN PROVEN.

This program is for example purposes only and not intended for a specific application.

Rung 3 contains a counter which is used to determine which MRX or MWX instruction to execute. Its purpose is to prevent multiple MRX/MWX rungs being active at the same time. Since the counter may only have one value at any particular time, only a single rung may be executed.

Please also note that adding additional MRX/MWX rungs would be accomplished simply by increasing the K4 value to the new total number of MRX and MWX instructions needed. SP116 is used to increment the counter so that each time an MRX or MWX is executed, the counter then enables the next MRX or MWX once the current MRX or MWX is complete.



MRX (rung 4)
Port Number: K2
Slave Address: K1
Function Code: 03 - Read Holding Registers
Start Slave Memory Address: 40001
Start Master Memory Address: V3000
Number of Elements: 12
Modbus Data type: 584/984 Mode
Exception Response Buffer: V405

MWX (rung 5)
Port Number: K2
Slave Address: K1
Function Code: 16 - Preset Multiple Registers
Start Slave Memory Address: 40012
Start Master Memory Address: V2000
Number of Elements: 4
Modbus Data type: 584/984 Mode
Exception Response Buffer: V400

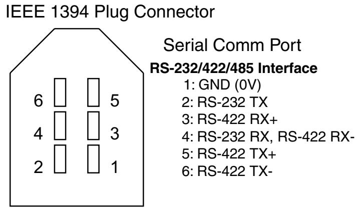
MRX (rung 6)
Port Number: K2
Slave Address: K2
Function Code: 03 - Read Holding Registers
Start Slave Memory Address: 40001
Start Master Memory Address: V3020
Number of Elements: 12
Modbus Data type: 584/984 Mode
Exception Response Buffer: V415

MWX (rung 7)
Port Number: K2
Slave Address: K2
Function Code: 16 - Preset Multiple Registers
Start Slave Memory Address: 40012
Start Master Memory Address: V2020
Number of Elements: 4
Modbus Data type: 584/984 Mode
Exception Response Buffer: V410

Communicating with Third-party Devices

The *Sure*Servo™ Serial Comm Port supports RS-232/422/485 communications. The drive can be set up to communicate on standard Modbus networks using ASCII or RTU transmission modes. Using the drive's Communication Protocol parameters, you can select the desired mode, data bits, parity, and stop bits. The communication parameters must be the same for all devices on a Modbus network.

Most drive parameters can be written to or updated from a master controller using Modbus communications. However, the drive's operational "run" commands (i.e Servo On, Command Trigger, RESET, etc) can only be executed by controlling the drive's physical digital inputs.



*Sure*Servo™ Block Transfer Function

A group of Status Monitor Registers (P0-04 to P0-08) and a group of Block Data Registers (P0-09 to P0-16) are available in the *Sure*Servo drive. These continuous block of registers can be used to "group" miscellaneous drive parameters together allowing you to read/write the desired parameters in one block instead of having to use a Read/Write command for each parameter.

P2-30 - setting this parameter to (5) will disable "parameter write to EEPROM" each time communications is attempted with the drive (default 0). This parameter setting is not retained when power is disconnected from the drive.

SureServo drives have a provision for shutting down control power to the output of the drive in the event of a communications timeout. This is set up using drive parameters P3-03 and P3-04, along with a digital output configured for servo fault alarm.

Common Modbus RTU Masters

- KEP*Direct* for PLCs (serial communications only)
- Think & Do Live 5.6, Studio 7.2.1 (serial communications only)
- MODSCAN from www.wintech.com

For additional technical assistance, go to our Technical support home page at:
http://support.automationdirect.com/technotes.html

Modbus Protocol Modes

This section explains the specifics of the Modbus protocols. It is not necessary to use this information if your drive control is capable of serving as a Modbus master controller.

ASCII Mode:

Each 8-bit data is the combination of two ASCII characters. For example, a 1-byte data: 64 Hex, shown as '64' in ASCII, consists of '6' (36Hex) and '4' (34Hex).

The following table shows the available hexadecimal characters and their corresponding ASCII codes.

Character	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'
ASCII Code	30H	31H	32H	33H	34H	35H	36H	37H
Character	'8'	'9'	'A'	'B'	'C'	'D'	'E'	'F'
ASCII Code	38H	38H	41H	42H	43H	44H	45H	46H

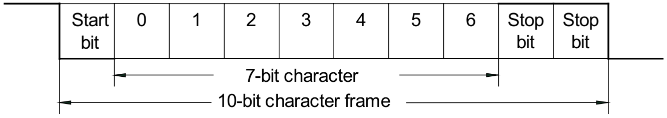
RTU Mode:

Each 8-bit data is the combination of two 4-bit hexadecimal characters. For example, a 1-byte data: 64 Hex.

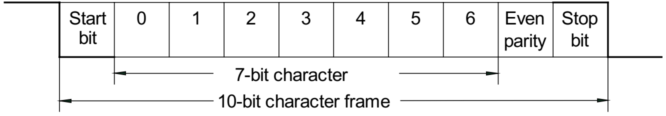
Modbus ASCII and RTU Data Format

10-bit character frame (For 7-bit character):

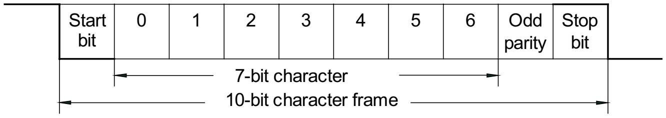
P3-02 = 00: ASCII mode (7 data bits, no parity, 2 stop bits)



P3-02 = 01: ASCII mode (7 data bits, even parity, 1 stop bit)

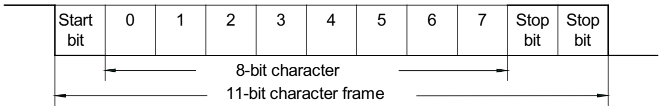


P3-02 = 02: ASCII mode (7 data bits, odd parity, 1 stop bit)

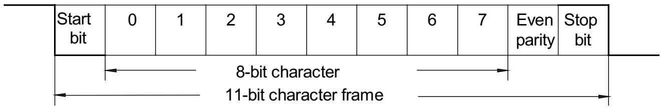


11-bit character frame (For 8-bit character):

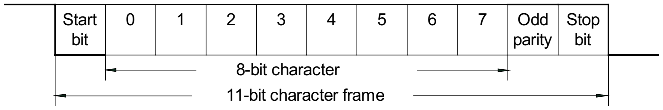
P3-02 = 03: ASCII mode (8 data bits, no parity, 2 stop bits)
P3-02 = 06: RTU mode (8 data bits, no parity, 2 stop bits)



P3-02 = 04: ASCII mode (8 data bits, even parity, 1 stop bit)
P3-02 = 07: RTU mode (8 data bits, even parity, 1 stop bit)



P3-02 = 05: ASCII mode (8 data bits, odd parity, 1 stop bit)
P3-02 = 08: RTU mode (8 data bits, odd parity, 1 stop bit)



Communication Protocol

Modbus ASCII Mode:

STX	Start Character: (3AH)
ADR 1	Communication Address: 8-bit address consists of 2 ASCII codes
ADR 0	
CMD 1	
CMD 0	
DATA (n-1)	Contents of data: n x 8-bit data consists of 2n ASCII codes. n[]25 maximum of 50 ASCII codes
....	
DATA 0	
LRC CHK 1	LRC check sum: 8-bit check sum consists of 2 ASCII codes
LRC CHK 0	
END 1	END characters: END 1=CR (0DH), END 0 =LF (0AH)
END-0	

Modbus RTU Mode:

START	A silent interval of more than 10 ms
ADR	Communication Address: 8-bit address
CMD	
DATA (n-1)	Contents of data: n x 8-bit data,n<=25
....	
DATA 0	
CRC CHK Low	CRC check sum: 16-bit check sum consists of 2 8-bit characters
CRC CHK High	
END	A silent interval of more than 10 ms

ADR (Communication Address)

Valid communication addresses are in the range of 0 to 254. A communication address equal to 0 means broadcast to all *Sure*Servo drives. In this case, the drive will not reply any message to the master device.

For example, communication to drive with address 16 decimal:

Modbus ASCII mode: (ADR 1, ADR 0)=’1’,’0’ => ‘1’=31H, ‘0’=30H
Modbus RTU mode: (ADR)=10H

CMD (Command) and DATA (data characters)

The format of data characters depends on the command code. The available command codes are described as follows: Command code: 03H, read N words. The maximum value of N is 10. For example, reading continuous 2 words from starting address 0200H of drive with address 01H.

Modbus ASCII mode:

Command Message	
STX	‘:’
ADR 1 ADR 0	‘0’
	‘1’
CMD 1 CMD 0	‘0’
	‘3’
Starting data address	‘0’
	‘2’
	‘0’
	‘0’
Number of data (Count by word)	‘0’
	‘0’
	‘0’
	‘2’
LRC CHK 1 LRC CHK 0	‘F’
	‘8’

END 1
END 0
	CR

	LF

Response Message	
STX ‘:’	‘:’
ADR 1 ADR 0	‘0’
	‘1’
CMD 1 CMD 0	‘0’
	‘3’
Number of data (Count by byte)	‘0’
	‘4’
Content of starting data address 0200H	‘0’
	‘0’
	‘B’
	‘1’
Content data address 0201H	‘1’
	‘F’
	‘4’
	‘0’
LRC CHK 1 LRC CHK 0	‘E’
	‘8’

END 1
END 0
	CR

	LF

Modbus RTU mode:

Command Message	
ADR	01H
CMD	03H
Starting data address	02H
	00H
Number of data (Count by word)	00H
	02H
CRC CHK Low CRC CHK High	C5H
	B3H

Response Message	
ADR	01H
CMD	03H
Number of data (Count by byte)	04H
	‘0’
Content of data address 0200H	00H
	B1H
Content of data address 0201H	1FH
	40H
CRC CHK Low CRC CHK High	A3H
	D4H

Command code: 06H, write 1 word

For example, writing 100(0064H) to address 0200H of drive with address 01H.

Modbus ASCII mode:

Command Message	
STX	‘:’
ADR 1	‘0’
ADR 0	‘1’
CMD 1	‘0’
CMD 0	‘6’
Data Address	‘0’
	‘2’
	‘0’
	‘0’
	‘0’
	‘0’
	‘6’
	‘4’
LRC CHK 1	‘9’
LRC CHK 0	‘3’
END 1	CR
END 0	LF

Response Message	
STX ‘:’	‘:’
ADR 1	‘0’
ADR 0	‘1’
CMD 1	‘0’
CMD 0	‘6’
Data Address	‘0’
	‘2’
	‘0’
	‘0’
Data Content	‘0’
	‘0’
	‘6’
	‘4’
	‘9’
	‘3’
END 1	CR
END 0	LF

Modbus RTU mode:

This is an example of using function code 16 for writing to multiple registers.

Command Message	
ADR	01H
CMD	10H
Starting data address	02H
	00H
Number of data (Count by byte)	04H
Content of data address 0200H	00H
	02H
Content of data address 0201H	02H
	58H
CRC CHK Low	CBH
CRC CHK High	34H

Response Message	
ADR	01H
CMD	10H
Starting data address	02H
	00H
Number of data (Count by word)	00H
	02H
CRC CHK Low	4AH
CRC CHK High	08H

CHK (CHECK SUM)

Modbus ASCII Mode:

LRC (Longitudinal Redundancy Check) is calculated by summing up module 256, the values of the bytes from ADR1 to last data character, then calculating the hexadecimal representation of the 2’s-complement negation of the sum.

For example, reading 1 word from address 0201H of the drive with address 01H.

Command Message	
STX	‘:’
ADR 1 ADR 0	‘0’
	‘1’
CMD 1 CMD 0	‘0’
	‘3’
Starting data address	‘0’
	‘2’
	‘0’
	‘1’
Number of data (Count by word)	‘0’
	‘0’
	‘0’
	‘1’
LRC CHK 1 LRC CHK 0	‘F’
	‘8’

END 1
END 0
	CR

	LF

01H+03H+02H+01H+00H+01H=08H,
the 2’s complement negation of 08H is F8H.

Modbus RTU Mode:

Response Message	
	01H
CMD	03H
Starting data address	02H
	01H
Number of data (Count by word)	00H
	02H
CRC CHK Low CRC CHK High	6FH
	F7H

CRC (Cyclical Redundancy Check) is calculated by the following steps:

1) Load a 16-bit register (called CRC register) with FFFFH.
2) Exclusive OR the first 8-bit byte of the command message with the low order byte of the 16-bit CRC register, putting the result in the CRC register.
3) Shift the CRC register one bit to the right with MSB zero filling. Extract and examine the LSB.
4) If the LSB of CRC register is 0, repeat step 3, else Exclusive or the CRC register with the polynomial value A001H.
5) Repeat step 3 and 4 until eight shifts have been performed. When this is done, a complete 8-bit byte will have been processed
6) Repeat steps 2 to 5 for the next 8-bit byte of the command message.

Continue doing this until all bytes have been processed. The final contents of the CRC register equal the CRC value.

When transmitting the CRC value in the message, the upper and lower bytes of the CRC value must be swapped, i.e. the lower order byte will be transmitted first.

The following is an example of CRC generation using C language. The function takes two arguments:

Unsigned char* data ← a pointer to the message buffer

Unsigned char length ← the quantity of bytes in the message buffer

The function returns the CRC value as a type of unsigned integer.

```
Unsigned int crc_chk(unsigned char* data, unsigned char length){
  int j;
  unsigned int reg_crc=0xFFFF;
  while(length--){
        reg_crc ^= *data++;
        for(j=0;j<8;j++){
          if(reg_crc & 0x01){  /* LSB(b0)=1 */
            reg_crc=(reg_crc>>1) ^ 0xA001;
        }else{
            reg_crc=reg_crc >>1;
        }
      }
    }
  return reg_crc;
}
```

Modbus RTU mode is preferred. Limited support is available to Modbus ASCII users.