SERIAL COMMUNICATIONS

NOTE: SR44 soft starters have been discontinued. Please consider SR55 soft starters as a replacements.

Contents of this Chapter...

WARNING: The owner, installer, and user are responsible for the correct installation and use of the SR44, and must ensure that only qualified personnel install the SR44. The owner, installer, and user must ensure that the installation, operation and maintenance of the SR44 complies with the relevant Codes of Practice, Regulations, and Statutory Requirements. The Manufacturer or his agent do not assume any liability, expressed or implied, for any consequence resulting from inappropriate, negligent, or incorrect installation, application, use or adjustment of the product or circuit design, or from the mismatch of the SR44 to a motor. To prevent an electrical shock hazard, the SR44 must be properly connected to a safety ground. The SR44 is not designed for use in hazardous areas. Use in such an area may invalidate the hazardous area certification.

6.1 – Overview

The SR44 has built-in serial communication capability that is a compatible subset of the widely recognized Modbus RTU protocol (slave), and the SR44 may be connected to a PC or suitable PLC network via an optional SR44-RS485 interface card.

Serial communication requires an optional interface card (SR44-RS485), which includes a "Local/Remote" switch to easily switch starter control from communication to keypad.
Serial communication is valid only in the "Remote" position, and the local keypad powers down.
In the "Local" position, the local keypad powers up and controls the soft starter, and all serial communication is ignored.

6.2 – Quick Start

Work should only be carried out on the SR44 with all electrical power disconnected

1) Communication settings:
   a) 9600 baud, 8 data bits, 1 stop bit, no parity.
   b) network station number = 1.
2) Before Connecting a Comm interface, use the standard keypad to make any changes for your bus system:
   a) Set Parameter 1 to the required network station number (slave address).
   b) If not continously polling the unit at least once every 5 seconds, set P-126 to 0, turning off comms timeout trip
   c) Make any changes to Parameter 121, bit 6.
      0 = 1 Stop bit.
      1 = 2 Stop bits
   d) Store changes to EEROM (save RAM).
3) The Local/Remote switch must be set to "Remote" to use the optional communication interface SR44-RS485. When this switch is set to "Local", serial communication is ignored and the integrated local keypad controls the soft starter. (Refer to "Chapter 7: Optional Hardware" for information about installing and using the optional Local/Remote switch.)
4) Register and coil addresses start at zero.

SR44 has a Modbus offset of one address for registers and bits. PLC programs should target one address higher than desired, e.g., PLC targets SR44 address 40019 to read/write to/from address 40018 (P-18).

Allowed address ranges are Register numbers 0…127, EEPROM Registers 1000…1127, Coils (flags) 0…1024.

(Refer to the "Parameter Memory Addresses" section of this chapter for more information.)

6.2 – Quick Start (continued)

5) Function 06 (Write Single Register)
   Parameters in the SR44 are 8-bit bytes which are written to by 16-bit Modbus words. Only the low byte of the Modbus word in the PLC gets sent to the SR44 parameter.
6) Function 05 (Write Single Coil)
   This function is used to set or clear a single bit within the Read/Write parameters between 0 through 127. (Only the bits within parameters designated as Read/Write can be preset with this function; Parameters 18, 51, 52, 53, 54, 86, 95, 96, 121, & 127)
7) Function 03 (Read Holding Registers)
   This function reads 8-bit bytes back to the PLC. If more than one byte is requested from the SR44, the PLC program will have to separate 16-bit Modbus words into the 8-bit SR44 bytes.
8) SR44 Status Bits (User Flags) are read as holding registers, i.e., read register 9 to read Status2 bits. For status bit writes, bits are mapped one-to-one, e.g., the Kickstart flag bit 0 of User Flags 1 (P51) is bit no 51 * 8 + 0 = bit 408.
   (Refer to the "Specifications" section of this chapter for supported Modbus function codes.)

6.3 – Specifications

1. Refer to Modicon Modbus Protocol Reference Guide PI-MBUS-300 at www.modbus.org for full protocol specification.
2. Baud rate is 9600.
3. 8 data bits, no parity, 1 stop (default) or 8 data bits, no parity, two stop bits.
4. Modbus timeout is 3.5 character times. The in message timeout is also 3.5 character times, unlike the standard which is 1.5 character times. Further, it is possible that the SR44 will be delayed in responding to a request if it is checking a fault condition.
5. Broadcast is not supported.
6. SR44 parameters retain their normal numbers (0-127).
7. Status bits (flags) are bit positions starting at parameter 0 bit 0.
8. Register numbers are 0…127.
9. Coils (flags) 0…1024.
10. Offline EEROM Registers are 1000…1127.
    (Non-volatile registers where the "Permanent Store" parameters are saved.)
11. Supports the following Modbus functions:

    Function 03 read holding registers up to 4 words (8 parameters) in number.

    Function 05 write single coil.

    Function 06 write single register (byte).

    Function 07 exception status.
    - Error 01 illegal function (write to read-only parameter).
    - Error 02 illegal data address (bad parameter number).
    - Error 03 illegal data (trying to read or write too much data).
    - Error 06 slave busy.
    - Error 07 negative acknowledge.

6.4 – Parameter Memory Addresses

Parameter Memory Addresses							
Parameter Number	Modbus *		Parameter Bit #s **			Name / Function	Type ***
	Address	Byte	Individual Bit	Decimal	Hex		
0	40000					dummy parameter	R
1	40001					Station No	R/W
2	40002					Language	R/W
3	40003			n/a		Program Type	R
4	40004					Software Ver	R
5	40005					**Password**	R/W
6	40006					Firing Mode	R/W
7	40007					Protection	R/W
8	40008	High	–	–	–	Status 1	R
			0 (1)	64	40	Stopped	
			1 (2)	65	41	Starting	
			2 (4)	66	42	Current Limit	
			3 (8)	67	43	Top of Ramp	
			4 (16)	68	44	Dwell	
			5 (32)	69	45	Full Conduction	
			6 (64)	70	46	Energy Saving	
			7 (128)	71	47	Stopping	
9	40009		–	–	–	Status 2	R
			0 (1)	72	48	Alarm	
			1 (2)	73	49	Overload Integrating	
			2 (4)	74	4A	Stall	
			3 (8)	75	4B	Forced Override	
			4 (16)	76	4C	Noise	
			5 (32)	77	4D	Inhibit Start	
			6 (64)	78	4E	DC I/p Level Reached	
			7 (128)	79	4F	Reserved	

** Modbus addresses apply only to Function 06 (Write Single Register), or to only the first parameter of Function 03 (Read Holding Registers).*

*** Parameter Bit #s apply only to Function 05 (Write Single Coil).*

**** Type: R = Read Only; R/W = Read/Write*

Table continued next page.

6.4 – Parameter Memory Addresses (continued)

Parameter Memory Addresses (continued)							
Parameter Number	Modbus *		Parameter Bit #s **			Name / Function	Type ***
	Address	Byte	Individual Bit	Decimal	Hex		
10	40010	High	–	–	–	Status 3	R
			0 (1)	80	50	Relay K1 Status	
			1 (2)	81	51	Relay K2 Status	
			2 (4)	82	52	(unused)	
			3 (8)	83	53	(unused)	
			4 (16)	84	54	Input 1 Status	
			5 (32)	85	55	(unused)	
			6 (64)	86	56	(unused)	
			7 (128)	87	57	(unused)	
11	40011		n/a			Start Ped	R/W
12	40012					Start Time	R/W
13	40013					Kick Ped	R/W
14	40014					Kick Time	R/W
15	40015					Dwell Time	R/W
16	40016					Stop Ped	R/W
17	40017					Stop Time	R/W
18	40018		–	–	–	Auto Config	R/W
			0 (1)	144	90	Auto Ramp	
			1 (2)	145	91	Auto 3MC	
			2 (4)	146	92	Auto Bypass	
			3 (8)	147	93	Auto Jog	
			4 (16)	148	94	Auto end stop	
			5 (32)	149	95	Auto stop	
			6 (64)	150	96	Auto end start	
			7 (128)	151	97	Auto Pedestal	
19	40019		n/a			Opt Rate	R/W
20	40020					Reference PF	R
21	40021					Internal PF	R
22	40022					Delay Angle	R
23	40023					Max Angle	R

** Modbus addresses apply only to Function 06 (Write Single Register), or to only the first parameter of Function 03 (Read Holding Registers).*

*** Parameter Bit #s apply only to Function 05 (Write Single Coil).*

**** Type: R = Read Only; R/W = Read/Write*

Table continued next page.

6.4 – Parameter Memory Addresses (continued)

Parameter Memory Addresses (continued)							
Parameter Number	Modbus *		Parameter Bit #s **			Name / Function	Type ***
	Address	Byte	Individual Bit	Decimal	Hex		
24	40024	High	n/a			Reserved – Rated Amps (100s)	R
25	40025					Reserved – Rated Amps	R
26	40026					Current	R
27	40027					Peak Start I	R
28	40028					Low I Level	R/W
29	40029					Low I Time	R/W
30	40030					C/L Level	R/W
31	40031					C/L Time	R/W
32	40032					Shear Level	R/W
33	40033					Shear Time	R/W
34	40034					Ovld Level	R/W
35	40035					Ovld Delay	R/W
36	40036					% Overload	R
37	40037					Reserved	–
38	40038		–	–	–	UserFlags 5 – Reserved	n/a
			0 (1)	304	130	(unused)	
			1 (2)	305	131	Reserved	
			2 (4)	306	132	Reserved	
			3 (8)	307	133	(unused)	
			4 (16)	308	134	(unused)	
			5 (32)	309	135	(unused)	
			6 (64)	310	136	(unused)	
			7 (128)	311	137	(unused)	
39	40039		n/a			PF Angle	R
40	40040					Supply Volts	R
41	40041					Reserved	–
42	40042					Reserved	–
43	40043					Reserved	–
44	40044					Reserved	–
45	40045					Reserved	–
46	40046					Reserved	–
47	40047					Reserved	–

** Modbus addresses apply only to Function 06 (Write Single Register), or to only the first parameter of Function 03 (Read Holding Registers).*

*** Parameter Bit #s apply only to Function 05 (Write Single Coil).*

**** Type: R = Read Only; R/W = Read/Write*

Table continued next page.

6.4 – Parameter Memory Addresses (continued)

Parameter Memory Addresses (continued)							
Parameter Number	Modbus *		Parameter Bit #s **			Name / Function	Type ***
	Address	Byte	Individual Bit	Decimal	Hex		
48	40048	High	n/a			Reserved	–
49	40049					Reserved	–
50	40050					Reserved	–
51	40051		–	–	–	User Flags 1	R/W
			0 (1)	408	198	Kickstart	
			1 (2)	409	199	Current Limit Timeout Sel'd	
			2 (4)	410	19A	Overload Selected	
			3 (8)	411	19B	Shearpin Selected	
			4 (16)	412	19C	Under Current Selected	
			5 (32)	413	19D	Thermistor Selected	
			6 (64)	–	–	(Unused)	
			7 (128)	415	19F	Terminal Starting	
52	40052		–	–	–	User Flags 2	R/W
			0 (1)	416	1A0	Low Voltage Soft-Stop	
			1 (2)	417	1A1	Not Global Enable 1	
			2 (4)	418	1A2	Not Global Enable 2	
			3 (8)	419	1A3	Pod Start/Stop	
			4 (16)	420	1A4	Board Start/Stop	
			5 (32)	421	1A5	Main Contactor	
			6 (64)	422	1A6	Second Parm Set	
			7 (128)	423	1A7	Zero Start Time	
53	40053		–	–	–	User Flags 3	R/W
			0 (1)	424	1A8	pf1 pol	
			1 (2)	425	1A9	pf2 pol	
			2 (4)	426	1AA	User Current Limit	
			3 (8)	427	1AB	User Trip	
			4 (16)	428	1AC	Invert Controller Input	
			5 (32)	429	1AD	Cooling (Inhibit Restart)	
			6 (64)	–	–	(Unused)	
			7 (128)	–	–	(Unused)	

**** Modbus addresses apply only to Function 06 (Write Single Register), or to only the first parameter of Function 03 (Read Holding Registers).***

***** Parameter Bit #s apply only to Function 05 (Write Single Coil).***

****** Type: R = Read Only; R/W = Read/Write***

Table continued next page.

6.4 – Parameter Memory Addresses (continued)

Parameter Memory Addresses (continued)							
Parameter Number	Modbus *		Parameter Bit #s **			Name / Function	Type ***
	Address	Byte	Individual Bit	Decimal	Hex		
54	40054	High	–	–	–	I/O Polarity	R/W
			0 (1)	432	1B0	K1 Polarity	
			1 (2)	433	1B1	K2 Polarity	
			2 (4)	–	–	(Unused)	
			3 (8)	–	–	(Unused)	
			4 (16)	436	1B4	I/p 1 Polarity	
			5 (32)	–	–	(Unused)	
			6 (64)	–	–	(Unused)	
			7 (128)	–	–	(Unused)	
55	40055		n/a			Reserved	–
56	40056					Reserved	–
57	40057					K1 Map	R/W
58	40058					K1 Bit Mask	R/W
59	40059					K2 Map	R/W
60	40060					K2 Bit Mask	R/W
61	40061					Reserved	–
62	40062					Reserved	–
63	40063					Reserved	–
64	40064					Reserved	–
65	40065					i/p1 Map	R/W
66	40066					i/p1 bit Msk	R/W
67	40067					Reserved	–
68	40068					Reserved	–
69	40069					Reserved	–
70	40070					Reserved	–
71	40071					Cntactor Dly	R/W
72	40072					Trip Snstvty	R/W
73	40073					Last Trip	R
74	40074					2nd Last Trp	R
75	40075					3rd Last Trp	R
76	40076					4thLast Trp	R
77	40077					5th Last Trp	R

** Modbus addresses apply only to Function 06 (Write Single Register), or to only the first parameter of Function 03 (Read Holding Registers).*

*** Parameter Bit #s apply only to Function 05 (Write Single Coil).*

**** Type: R = Read Only; R/W = Read/Write*

Table continued next page.

6.4 – Parameter Memory Addresses (continued)

Parameter Memory Addresses (continued)							
Parameter Number	Modbus *		Parameter Bit #s **			Name / Function	Type ***
	Address	Byte	Individual Bit	Decimal	Hex		
78	40078					Protection 2	R/W
79	40079					Start Ped 2	R/W
80	40080					Start Time 2	R/W
81	40081			n/a		Kick Ped 2	R/W
82	40082					Kick Time 2	R/W
83	40083					Dwell Time 2	R/W
84	40084					Stop ped 2	R/W
85	40085					Stop Time 2	R/W
86	40086	High	–	–	–	AutoConfig 2	R/W
			0 (1)	688	2B0	Auto Ramp	
			1 (2)	689	2B1	Auto 3MC	
			2 (4)	690	2B2	Auto Bypass	
			3 (8)	691	2B3	Auto Jog	
			4 (16)	692	2B4	Auto Endstop	
			5 (32)	693	2B5	Auto Stop	
			6 (64)	694	2B6	Auto Endstart	
			7 (128)	695	2B7	Auto Pedestal	
87	40087					Low I 2	R/W
88	40088					Low I time 2	R/W
89	40089					C/L 2	R/W
90	40090			n/a		C/L Time 2	R/W
91	40091					Shearpin 2	R/W
92	40092					Shear time 2	R/W
93	40093					Ovld Level 2	R/W
94	40094					Ovld Delay 2	R/W

** Modbus addresses apply only to Function 06 (Write Single Register), or to only the first parameter of Function 03 (Read Holding Registers).*

*** Parameter Bit #s apply only to Function 05 (Write Single Coil).*

**** Type: R = Read Only; R/W = Read/Write*

Table continued next page.

6.4 – Parameter Memory Addresses (continued)

Parameter Memory Addresses (continued)							
Parameter Number	Modbus *		Parameter Bit #s **			Name / Function	Type ***
	Address	Byte	Individual Bit	Decimal	Hex		
95	40095	High	–	–	–	User Flags12	R/W
			0 (1)	760	2F8	Kickstart (2)	
			1 (2)	761	2F9	Current Limit Timeout Sel'd (2)	
			2 (4)	762	2FA	Overload Selected (2)	
			3 (8)	763	2FB	Shearpin Selected (2)	
			4 (16)	764	2FC	Under Current Selected (2)	
			5 (32)	765	2FD	Thermistor Selected (2)	
			6 (64)	–	–	(Unused)	
			7 (128)	767	2FF	Terminal Starting (2)	
96	40096		–	–	–	User Flags22	R/W
			0 (1)	768	300	Low Voltage Soft-Stop (2)	
			1 (2)	769	301	Not Global Enable 1 (2)	
			2 (4)	770	302	Not Global Enable 2 (2)	
			3 (8)	771	303	Pod Start/Stop (2)	
			4 (16)	772	304	Board Start/Stop (2)	
			5 (32)	773	305	Main Contactor (2)	
			6 (64)	774	306	Second Parm Set (2)	
			7 (128)	775	307	Zero Start Time (2)	
97	40097		n/a			U1 I/P srce	R/W
98	40098					U1 I/P Mask	R/W
99	40099					U1 O/P Dest	R/W
100	40100					U1 O/P Mask	R/W
101	40101					U2 I/P srce	R/W
102	40102					U2 I/P Mask	R/W
103	40103					U2 O/P Dest	R/W
104	40104					U2 O/P Mask	R/W
105	40105					Reserved	–
106	40106					Reserved	–
107	40107					Reserved	–
108	40108					Reserved	–

** Modbus addresses apply only to Function 06 (Write Single Register), or to only the first parameter of Function 03 (Read Holding Registers).*

*** Parameter Bit #s apply only to Function 05 (Write Single Coil).*

**** Type: R = Read Only; R/W = Read/Write*

Table continued next page.

6.4 – Parameter Memory Addresses (continued)

Parameter Memory Addresses (continued)							
Parameter Number	Modbus *		Parameter Bit #s **			Name / Function	Type ***
	Address	Byte	Individual Bit	Decimal	Hex		
109	40109	High	n/a			Bypass Upper Limit	R/W
110	40110					Bypass Lower Limit	R/W
111	40111					Reserved	–
112	40112		–	–	–	Status 4	R
			0 (1)	896	380	60 Hz	
			1 (2)	897	380	Phase Rotation Sequence	
			2 (4)	898	382	Shearpin	
			3 (8)	899	383	Under Current	
			4 (16)	900	384	Thermistor	
			5 (32)	901	385	Timeout	
			6 (64)	902	386	Overload	
			7 (128)	903	387	Thermal Switch	
113	40113		n/a			Reserved	–
114	40114					StopProfile	R/W
115	40115					Last App	R/W
116	40116					CoolingTime	R/W
117	40117					Reserved	–
118	40118					System Use Only	–
119	40119					Reserved	–
120	40120		–	–	–	Status 5	R
			0 (1)	960	420	Over Temperature	
			1 (2)	961	421	Offline Command Fail	
			2 (4)	962	422	(unused)	
			3 (8)	963	423	Fan Status	
			4 (16)	964	424	(unused)	
			5 (32)	965	425	(unused)	
			6 (64)	966	426	(unused)	
			7 (128)	967	427	(unused)	

** Modbus addresses apply only to Function 06 (Write Single Register), or to only the first parameter of Function 03 (Read Holding Registers).*

*** Parameter Bit #s apply only to Function 05 (Write Single Coil).*

**** Type: R = Read Only; R/W = Read/Write*

Table continued next page.

6.4 – Parameter Memory Addresses (continued)

Parameter Memory Addresses (continued)							
Parameter Number	Modbus *		Parameter Bit #s **			Name / Function	Type ***
	Address	Byte	Individual Bit	Decimal	Hex		
121	40121	High	–	–	–	UserFlags 4	R/W
			0 (1)	–	–	Impact Load	
			1 (2)	969	429	(unused)	
			2 (4)	970	42A	Auto Soft Stop Smoothing	
			3 (8)	971	42B	Breaker Present	
			4 (16)	–	–	Optimise +	
			5 (32)	973	42D	Quick Ramp Times	
			6 (64)	974	42E	Two Stop Bits	
			7 (128)	–	–	Auto Stop (2)	
122	40122		n/a			Stop Smooth	R/W
123	40123					Temp/Alt Derate	R/W
124	40124					OffLine Command	R/W
125	40125					Bus BaudRate	R/W
126	40126					Comms Trip	R/W
127	40127		–	–	–	Trip Flags	R/W
			0 (1)	1016	458	Trip 1 Enabled – Phase Loss	
			1 (2)	1017	459	Trip 8 Enabled – SCR Loss Motor (Motor-Side Thyristor Loss)	
			2 (4)	1018	45A	Trip 9 Enabled – Sensing Signal (Thyristor Sensing Trip Type A)	
			3 (8)	1019	45B	Trip 10 Enabled – SCR Shorted (Thyristor Sensing Trip Type B)	
			4 (16)	1020	45C	Trip 4 Enabled – SCR Firing (Thyristor Sensing Trip Type C)	
			5 (32)	1021	45D	Trip 5 Enabled – SCR Signal (Edge Sensing Trip)	
			6 (64)	1022	45E	Trip 6 Enabled – SCR Signal (Sensing Trip Type A)	
			7 (128)	1023	45F	Trip 7 Enabled – Sensing Signal (Sensing Trip Type B)	

* ***Modbus addresses apply only to Function 06 (Write Single Register), or to only the first parameter of Function 03 (Read Holding Registers).***

** ***Parameter Bit #s apply only to Function 05 (Write Single Coil).***

*** ***Type: R = Read Only; R/W = Read/Write***

6.5 – Serial Communication to SR44

SR44 parameters are stored in 8-bit bytes. Modbus reads the values from the slave as 16-bit words. Therefore two SR44 parameter bytes are packed into each Modbus word. The lowest parameter is in the data High of the word, and the next parameter is in the data Low. This also means that even when the read request is for one register, two SR44 parameters are read. Up to four contiguous words (8 parameters) may be read with one query.

SR44 has a Modbus offset of one address for registers and bits. PLC programs should target one address higher than desired, e.g., PLC targets SR44 address 40019 to read/write to/from address 40018 (P-18).

6.5.0 – Parameter Scaling

Certain parameters have non-unity scaling factors which must be applied when using serial communication to read and write the parameters. (Parameter scaling factors are shown in Chapter 5: Keypad Menu Items and Parameters.)

- Scaling applies only when using serial communication to read and write parameters.
- Scaling Factor = Maximum Parameter Value ÷ 255
- When READING from parameters, MULTIPLY the serial comm value by the scaling factor.
- When WRITING to parameters, DIVIDE your desired value by the scaling factor (multiply by the inverse of the scaling factor).

Parameter Scaling Example:

P-71 = Contactor Delay; Default value = 160 ms; Scaling Factor = 4

For default value = 160 ms:
Keypad shows “160 ms”; Comms data reads “40” [multiply by 4 to get actual value of 160]

To change value to 200 ms:
Enter “200” through keypad, or
Write “50” through serial comm [divide by 4 to determine value to write]
Keypad display will show “200 ms”; serial comm will read “50” [multiply by 4 to get actual value]

To change value to 240 ms:
Enter “240” through keypad, or
Write “60” through serial comm [divide by 4 to determine value to write]
Keypad display will show “240 ms”; serial comm will read “60” [multiply by 4 to get actual value]

The minimum value that a scaled parameter can be changed is equal to that parameter’s scaling factor.

Parameters referenced in the following examples:

Modbus Function 03 example 6.5.1:
- P-73 = Last Trip
- P-74 = 2nd Last Trip
- P-75 = 3rd Last Trip
- P-76 = 4th Last Trip
- P-77 = 5th Last Trip
- P-78 = Protection 2

Modbus Function 05 example 6.5.2:
- P-52 / B-6 = User Flags 2 / 2nd parameter set

Modbus Function 06 example 6.5.3:
- P-124 = OffLine Command

6.5.1 – Reading Holding Registers (Modbus Function 03)

This function is used to read a group of up to eight parameters from the SR44. The allowable address range for this function is 0-127 for working parameters and 1000–1127 for offline EEROM parameters.

Example – Modbus Function 03:
Read the fault history (P-73 – P-77) from SR44 slave #1 into a PLC

- Start Slave Address (SR44): 40074 *
- Start Master Address (PLC): internal PLC register 1
- Number of Elements: 3

SR44					PLC	
Parameter	Address / Byte *	Data Value			Hex Data	Address ***
		Decimal	Hex			
P-73 (Last Trip)	40073 / high	0 (no trip)	00		00 02	internal PLC register #1
P-74 (2nd Last Trip)	40073 / low	2 (too hot)	02	→		
P-75 (3rd Last Trip)	40074 / high	13 (overload)	0D		0D 0E	internal PLC register #2
P-76 (4th Last Trip)	40074 / low	14 (shearpin)	0E			
P-77 (5th Last Trip)	40075 / high	16 (external trip)	10		10 C0	internal PLC register #3
P-78 (Protection2) **	40075 / low **	192 (full+optimize)	C0			

**** SR44 Modbus addresses are variable.***
Only the 1st address of the Function 03 read instruction 'matches' the parameter number, but there is an offset of one address from the parameter#. If the PLC targets 40074, the PLC will return data starting with P-73. P-73 will be in the high byte, and P-74 will be in the low byte.

***** The Function 03 mutliple read instruction gets data from an even number of parameters. Unwanted data can be discarded by the PLC.***

****** PLC addresses are any valid user-assigned memory addresses.***

Modbus Function 03 Transaction Table			
Query		Response	
Field	Hex Byte	Field	Hex Byte
Slave Address	01	Slave Address	01
Function	03	Function	03
Start Address High	00	Byte Count	04
Start Address Low *	4A (decimal = 74) *	Data High Byte	00 (P-73)
No. of Registers High	00	Data Low Byte	02 (P-74)
No. of Registers Low	02	Data High Byte	0D (P-75)
CRC Low	PLC & SR44 automatically calculate CRC checksum values	Data Low Byte	0E (P-76)
CRC High		Data High Byte	10 (P-77)
		Data Low Byte	C0 (P-78)
		CRC Low Byte	PLC & SR44 automatically calculate CRC checksum values
		CRC High Byte	

**** The SR44 has a Modbus offset of one address for registers & bits; PLC targets one address higher.***

6.5.2 – Writing Single Coil (Modbus Function 05)

This function is used to set or clear a single bit within the Read/Write parameters between 0 through 127. (Only the bits within parameters designated as Read/Write can be preset with this function; Parameters 18, 51, 52, 53, 54, 86, 95, 96, 121, & 127)

Example – Modbus Function 05:
PLC writes "Select 2nd Parameter Group" to SR44 (send a value of 1 to Bit#6 of P-52):

- Start Slave Bit Number (SR44): 423*
- Start Master Address (PLC): internal PLC bit address #1 (value = 1)

SR44					PLC	
Parameter Bit	Address *	Data Value		←	Hex Data	Address **
		Decimal	Hex			
P-52 / Bit 06 Second Parm Set	423 *	1	1		1	internal PLC bit address #1

**** SR44 bit addresses have an offset of one address from the parameter bit#. The PLC needs to target slave bit #423 in order to change bit #422.***

***** PLC addresses are any valid user-assigned bit addresses.***

Modbus Function 05 Transaction Table			
Query		Response	
Field	Hex Byte	Field	Hex Byte
Slave Address	01	Slave Address	01
Function	05	Function	05
Bit Address	1A7 (decimal = 423) *	Bit Address	1A7 (bit 422*) (bit 06 of P-52)
Force Data	01 (select 2nd parameter set)	Force Data	01 (select 2nd parameter set)
CRC Low	PLC & SR44 automatically calculate CRC checksum values	CRC Low Byte	PLC & SR44 automatically calculate CRC checksum values
CRC High		CRC High Byte	

**** SR44 has a Modbus offset of one address for registers & bits; PLC targets one address higher.***

6.5.3 – Writing Single Register (Modbus Function 06)

This function is used to set the value of a single parameter. It differs from the Modbus protocol in that only the data low byte of the query is written to the parameter address. The data high byte is ignored. (The normal response for correct operation is an echo of the data.)

If the PLC uses Function 06 to write one word of data into the SR44, only the low byte of the PLC data is used. The PLC low byte is written into the SR44 parameter, and the PLC high byte is ignored.

Example – Modbus Function 06:
PLC writes "Enable" to SR44 (send a value of 2 to P-124):

- Start Slave Address (SR44): 40125 *
- Start Master Address (PLC): internal PLC register #1 (value = 2)

SR44					PLC	
Parameter	Address / Byte *	Data Value			Hex Data**	Address ***
		Decimal	Hex	←		
P-124 (OffLine Command)	40125* / high	02	02**		00 02**	internal PLC register #1

** SR44 Modbus addresses have an offset of one address from the parameter #. The PLC needs to target slave address 40125 in order to change P-124.*

*** For Function 06 single write instructions, the SR44 accepts only the low byte of the PLC word; the high byte is ignored. In this example, only P-124 is changed by the write instruction.*

**** PLC addresses are any valid user-assigned memory addresses.*

Modbus Function 06 Transaction Table			
Query		Response	
Field	Hex Byte	Field	Hex Byte
Slave Address	01	Slave Address	01
Function	06	Function	06
Address High	00	Address High	00
Address Low	7D (decimal = 125) *	Address Low	7D (P-124) *
Force Data High	n/a (doesn't matter)	Force Data High	n/a (ignored)
Force Data Low	02 (Enable Starter)	Force Data Low	02 (Enable Starter)
CRC Low	PLC & SR44 automatically calculate CRC checksum values	CRC Low	PLC & SR44 automatically calculate CRC checksum values
CRC High		CRC High	

** SR44 has a Modbus offset of one address for registers & bits; PLC targets one address higher.*

6.5.4 – P-124 Offline Commands

The SR44 has a mechanism for carrying out common functions, or offline ones which take a long time and require the SR44 to be off and disabled. Commands are carried out by writing various values to a Command Register (parameter 124). For example, writing a 1 to P-124 using function 06 disables the starter.

Once a value has been written to this special register, further Modbus queries will receive a "slave busy" error until the command execution is completed. If an error occurred during the execution of the command (say the unit was not disabled before execution), then the Offline Command failed flag will be set. If function 7 has been used to poll the SR44, then once the busy period is over this flag is available in the returned status byte position 80h.

Parameter 124 Command Codes *			
Value	**Command**	**Active**	**Meaning**
1	Disable Starter	Always	Stop the SR44 responding to start commands, remote or local. If running the SR44 will stop immediately. The status of any start signal is not changed.
2	Enable Starter	Always	The SR44 will respond to new or existing start commands.
3	Bus Starting	Always	The SR44 responds to start requests from the keypad / bus port.
4	Remote Starting	Always	The SR44 responds to start requests from the hardware input.
5	Reset Starter	Always	The SR44 is forced to a cold start.
6	Reset Trip	Always	Resets the SR44 to stopped and ready state.
7	Bus Start	When Enabled	If Starting = bus & unit is enabled, SR44 will start.
8	Bus Stop	When Enabled	SR44 will always stop.
9	Factory Default	When Disabled	The working RAM is loaded with factory default values. his may take up to 5 seconds.
10	Power On Default	When Disabled	The working RAM is loaded with the permanent store values.
11	Save RAM	When Disabled	The working RAM is saved to the permanent store for future use. This may take up to 5 seconds.
**** Use only Function Code 06 for P-124 Offline Commands.***			

6.5.5 – Read Exception Status (Modbus Function 07) – Starter 01

This function's format is per the Modbus standard, but the bits in the value returned have different meanings shown below. This is the quickest of the Modbus functions, and should be used in the context of the SR44 to check if a station is present or check if it is busy.

Modbus Function 07 Transaction Table			
Query		Response	
Field	**Hex Byte**	**Field**	**Hex Byte**
Slave Address	01	Slave Address	01
Function	07	Function	07
CRC Low	41	Exception Data	00
CRC High	E2	CRC Low	PLC & SR44 automatically calculate CRC checksum values
		CRC High	
Returned Exception Data			
Bit		**Meaning**	
0		Enabled	
1		Stopped	
2		Starting	
3		Current Limiting	
4		TOR (Start Complete)	
5		Cooling	
6		Alarm	
7		Offline Command Failed	

6.5.6 – Returned Errors

If there is a communication error in the query, the SR44 does not reply.

If the network communication is OK, but for some reason the SR44 cannot correctly execute the query, then it will return an error reply. In an error reply, the function byte is returned OR'ed with 80h, and the following byte contains the error number, e.g. if Function 07 cannot be executed, 87h will be returned.

Modbus Function 07 Transaction Table Showing Error Response			
Query		Response	
Field	**Hex Byte**	**Field**	**Hex Byte**
Slave Address	01	Slave Address	01
Function	**07**	Function	**87**
CRC Low	41	Exception Data	01
CRC High	E2	CRC Low	PLC & SR44 automatically calculate CRC checksum values
		CRC High	
Returned Error Response			
Number	**Modbus**	**Meaning**	
0	Illegal Function	Writing	
1	Illegal Data Address	Out of allowed parameter range	
2	Illegal Data	Too many data bytes in query	
3	Slave Busy	Slave is performing a long command	

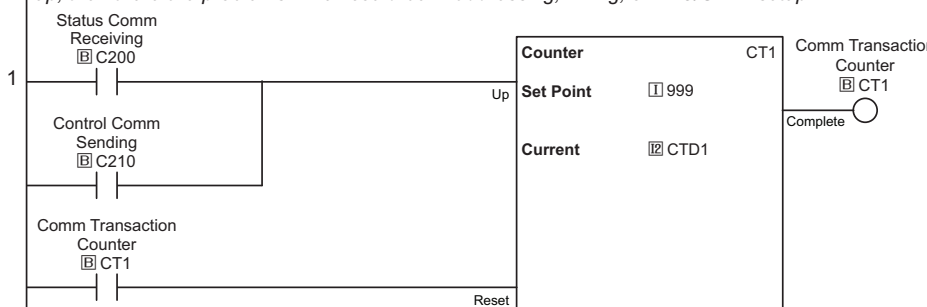
6.7 – Example PLC Communications Programs

6.7.1 – CLICK PLC Example Program

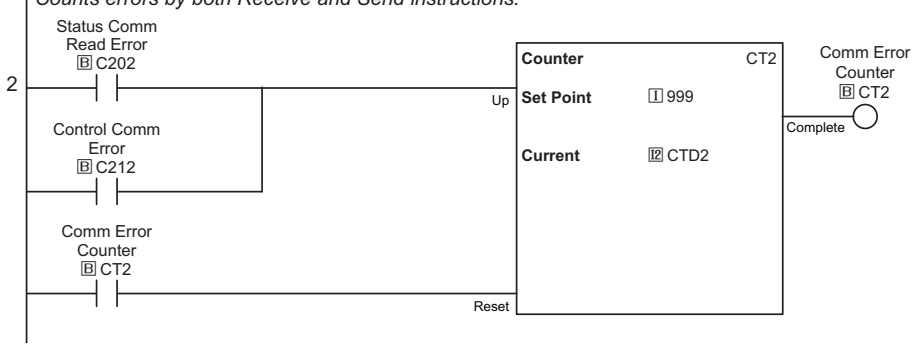
This information provided by AutomationDirect.com Technical Support is provided "as is" without a guarantee of any kind. These documents are provided by our Technical Support Department to assist others. We do not guarantee that the data is suitable for your application, nor do we assume any responsibility for them in your application.

This program is for illustrational purposes only, and is not intended for a true application.

The first 2 rungs monitor comm attempts and errors for troubleshooting. Counter CT1 counts the attempts by both Receive and Send instructions, and CT2 counts errors. CT1 will normally be counting rapidly, while CT2 will be counting very infrequently, if at all. If both counters are counting up, then there are problems which could be in addressing, wiring, or PLC/SR44 setup.

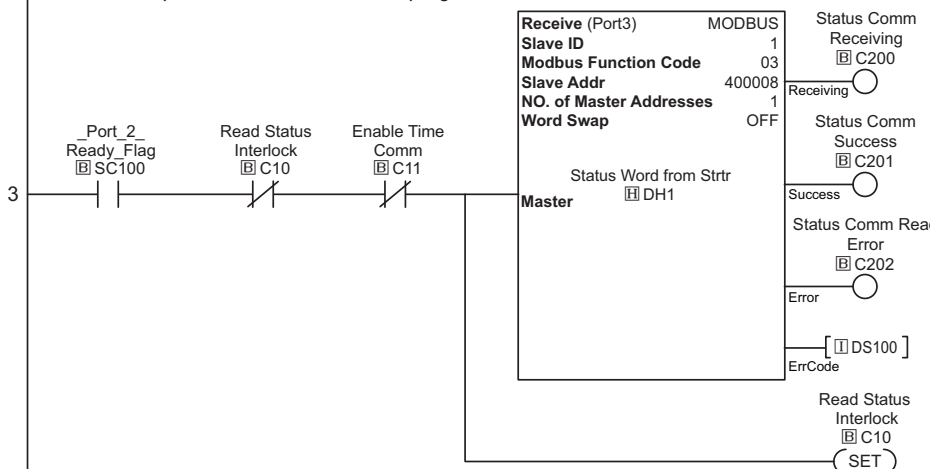


Counts errors by both Receive and Send instructions.

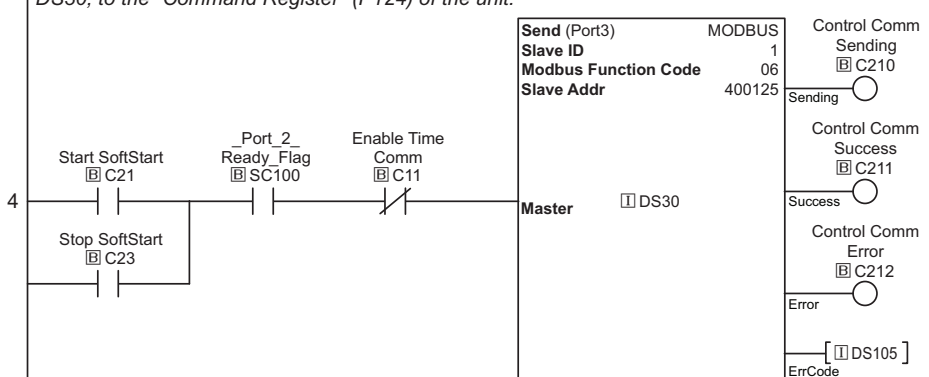


6.7.1 – CLICK PLC Example Program (continued)

This rung will be performing nearly all the comms with the starter. It will continuously read parameters P8 and P9 (the soft starter is 8-bit based) and the CLICK performs communications by 16-bit words. So the single value that is retrieved from the soft starter contains 2 values. Since both values are actually bit patterns, they are placed directly into a Hex data register DH1. This will be copied to C bits further in the program.



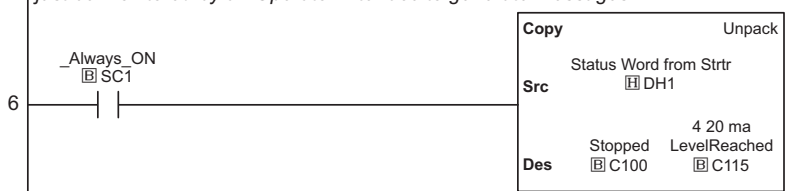
This rung will be used for Start/Stop control of the unit. It will only be activated when the user activates the Start and Stop inputs, which will send a value of "7" (Start) or "8" (Stop), contained in DS30, to the "Command Register" (P124) of the unit.



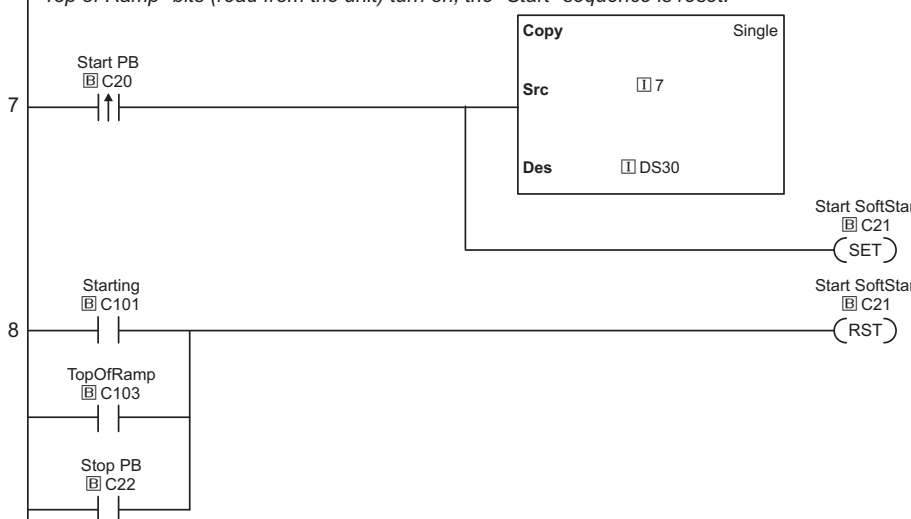
6.7.1 – CLICK PLC Example Program (continued)

This rung controls the interlocking of the 2 previous comm instructions. Since the Receive instruction is programmed to read continuously, this logic resets the interlock, UNLESS, the CLICK program calls for the Start/Stop Send instruction to be activated.

This rung takes the Status Word that was read from the Soft Start, and converts the word to control bits. The bits, C100 thru C115, could be used in your program to turn on outputs, and/or they could just be monitored by an Operator Interface to generate messages.

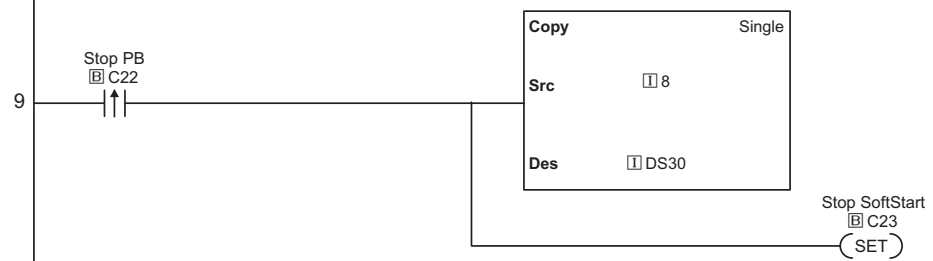


When "Start PB" is pressed (activated), a value of "7" is copied to DS30, and C21 is set. This will activate the Send instruction, which will write the contents of DS30 to P124. When the "Starting" or "Top of Ramp" bits (read from the unit) turn on, the "Start" sequence is reset.

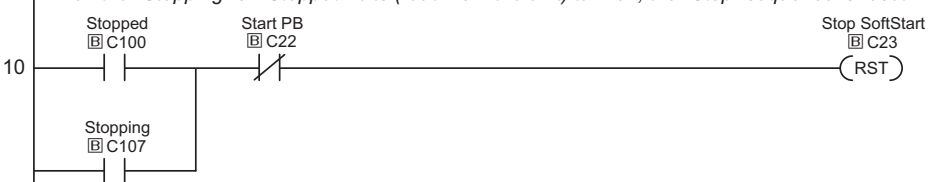


6.7.1 – CLICK PLC Example Program (continued)

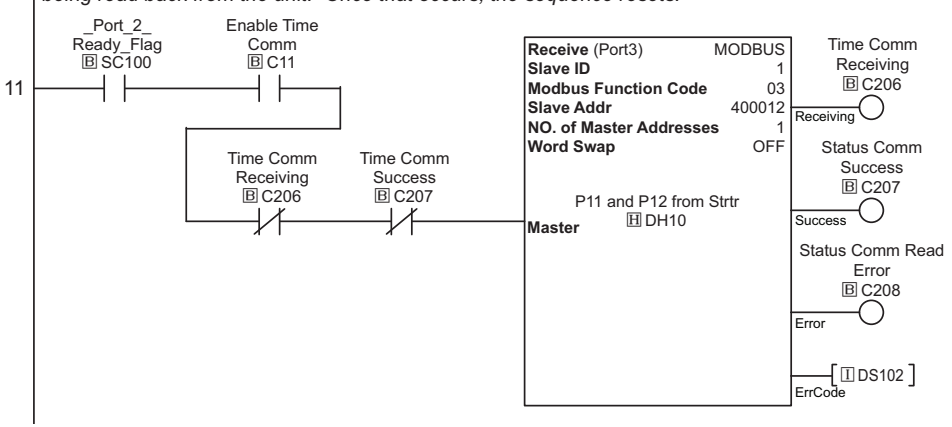
When "Stop PB" C22 is pressed (activated), a value of "8" (Stop) is copied to DS30, and C23 is set. This will activate the Send instruction, which will write the contents of DS30 to P124.



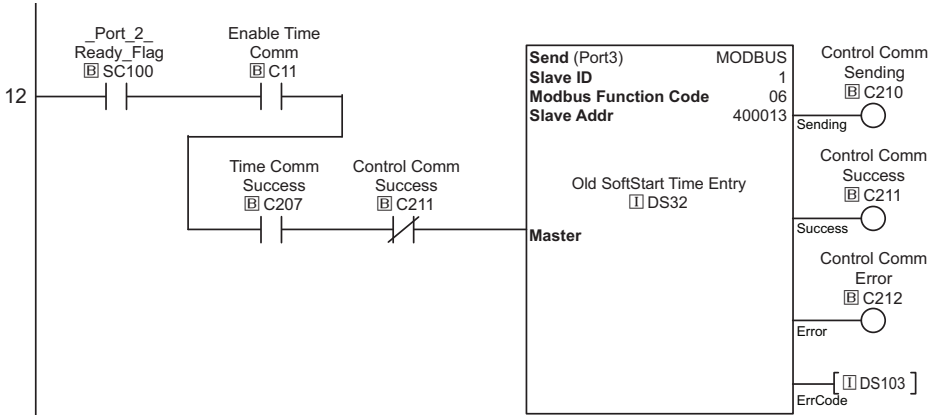
When the "Stopping" or "Stopped" bits (read from the unit) turn on, the "Stop" sequence is reset.



The next 2 rungs Receive and Send P12 (Soft Start Time) to the unit. This is complex because 2 hex values are actually retrieved (P11 and P12) by reading 1 Modbus address. The 2 rungs will synch themselves and communicate until the value that is sent to the unit is also confirmed by being read back from the unit. Once that occurs, the sequence resets.

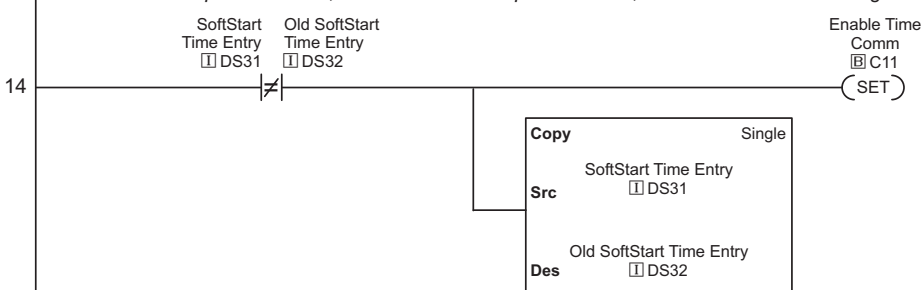


6.7.1 – CLICK PLC Example Program (continued)

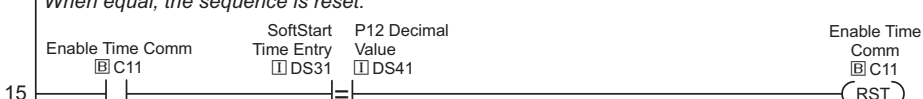


When both comm instructions have successfully operated, this rung will reset the communication bits and allow the comms to restart.

DS31 is assumed to come from an HMI (C-more for example), and when a new value is entered, it will not be equal to the old value, and this difference will set C11. That will start the Receive/Send sequence. Then, the new value is copied to DS32, which will turn off the rung.



This rung will be activated, until DS31 is equal to DS41, which is the Received value from the unit. When equal, the sequence is reset.



6.7.1 – CLICK PLC Example Program (continued)

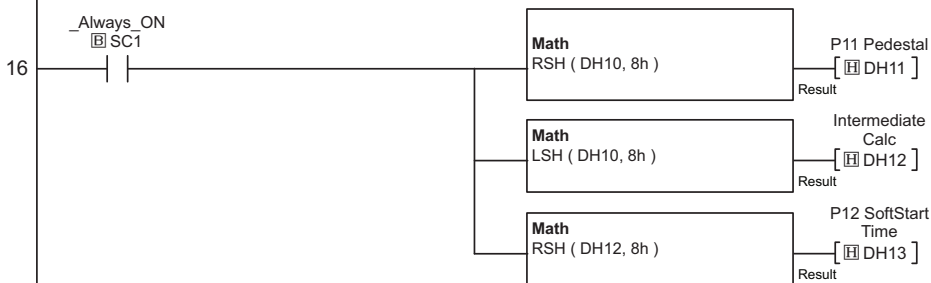
This rung manipulates the data retrieved (DH10) from the soft starter paramters P11 and P12. Since each Modbus word actually has 2 values (1 in each byte), the data has to be manipulated to put each relevant value in its' own register.

First, DH10 is loaded and the RSH instruction is performed to Shift the value to the right by 8 bits, which moves the upper byte to the lower byte, and this is output to DH11 (P11 Pedestal).

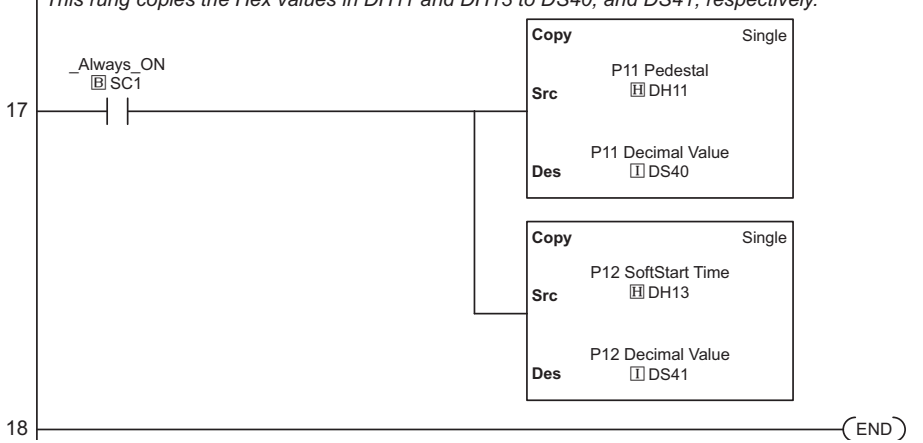
Next, the LSH (Left Shift Register) instruction is used in the 2nd MATH box to reload the original value in DH10, and move the lower byte to the upper byte (shifting left 8 bits), and we put the result in DH12 as an intermediate step.

Finally, we load DH12 in the 3rd MATH box, and perform a RSH (Right Shift Register) instruction to move the upper byte back to the lower byte, again, shifting 8 bits. Now the result is the bare lower byte value from the original word and this result is output to DH13 as our final value.

To summarize, the original data is shifted to the right 8 bits, which moved our original upper byte to the lower byte, and zeroed out the upper byte. This was the 1st MATH box. Then, the original value is reloaded, moved to the left 8 bits, then moved to the right 8 bits. This zeroed out the upper byte and resulted in the lower byte value. This took 2 MATH boxes. The values now reside in DH11 and DH13.



This rung copies the Hex values in DH11 and DH13 to DS40, and DS41, respectively.



6.7.2 – *Direct*LOGIC PLC Example Program

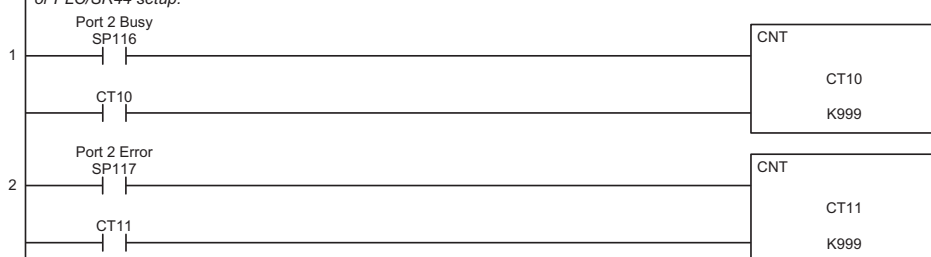
This information provided by AutomationDirect.com Technical Support is provided "as is" without a guarantee of any kind. These documents are provided by our Technical Support Department to assist others. We do not guarantee that the data is suitable for your application, nor do we assume any responsibility for them in your application.

This *Direct*LOGIC ladder program example works with the MWX and MRX instructions found in the DL06 and DL260 PLC platforms.

(The DL06 and the DL260 can do RS-485 directly out of port 2.)

This program is for illustrational purposes only, and is not intended for a true application.

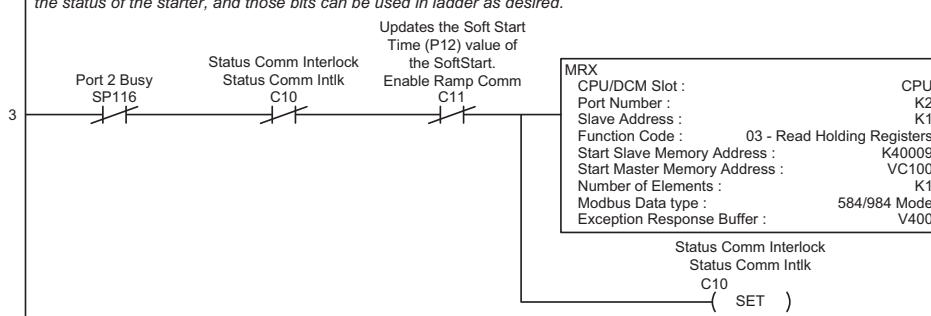
The first 2 rungs monitor comm attempts and errors for troubleshooting. Counter CT0 counts the attempts by both MRX and MWX instructions, and CT1 counts errors. CT0 will normally be counting rapidly, while CT1 will be counting very infrequently, if at all. If both counters are counting up, then there are problems which could be in addressing, wiring, or PLC/SR44 setup.



This rung will be doing nearly all the Comms with the soft starter.

It reads the Status Bits from the soft starter, from P8 and P9. Note that 8 is added for the starting address of 40001, 40001 + 8 (P8's address), and the result is 40009 for the "Start Slave Memory Address" field in the MRX.

P8 and P9 are read as one 16-bit word, and they are stored as a word starting at C100, so the bits C100-C117 will reflect the status of the starter, and those bits can be used in ladder as desired.

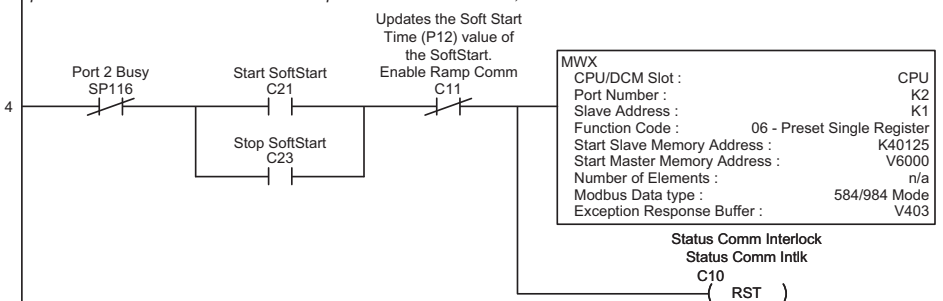


6.7.2 – *Direct*LOGIC PLC Example Program (continued)

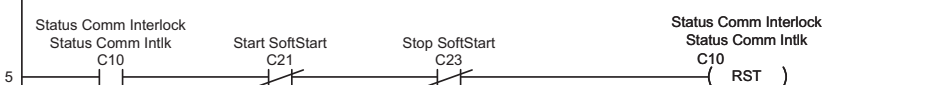
This rung will be used when the user activates it. Ordinarily, it will only be active for a short period because the user is Starting or Stopping the soft starter.

This rung will write the value contained in V6000 to the Command Register (P124). The values of 7 and 8 will "Start" and "Stop" the soft starter, respectively. Note that 124 is added to the starting address of 40001, and we have 40125 for the "Start Slave Memory Address".

C21 (Starting) and C23 (Stopping) are located lower in the ladder, and they will enable Comms to the soft starter until the pertinent Status bits relfect the desired operation of the soft starter, then those Status bits will reset C21 and C23.

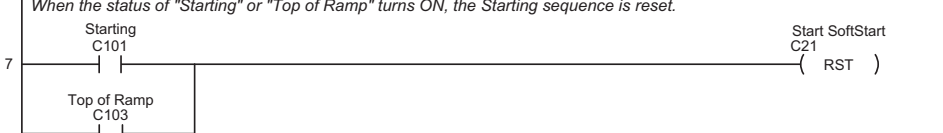


This rung will reset the Status Comm Interlock if no attempt is made to write to the soft starter. This will allow rung 3 to read the status again. Status will be read continuously from the starter, and writes will be done by excepton, that is, when the user causes it.

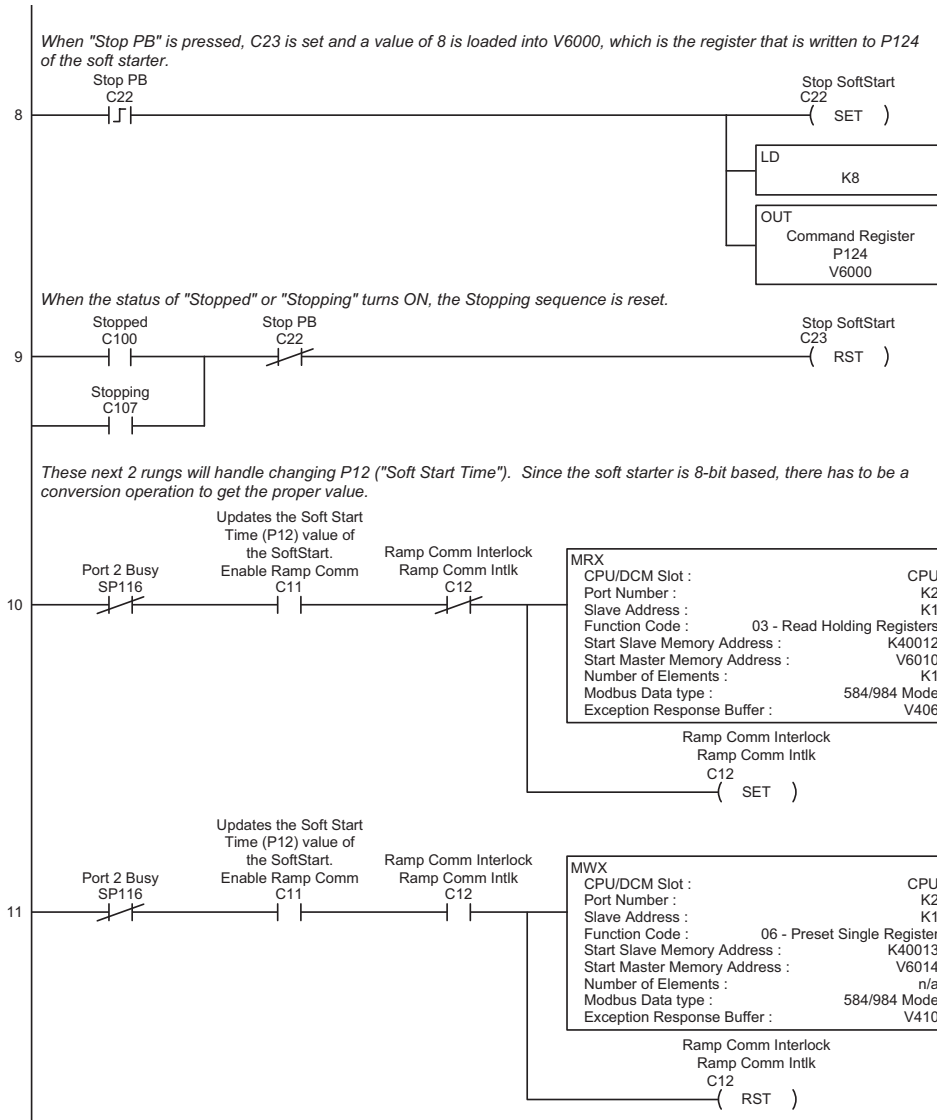


When "Start PB" is pressed, C21 is set, and a value of 7 is loaded into V6000, which is the register that is written to P124 of the soft starter.

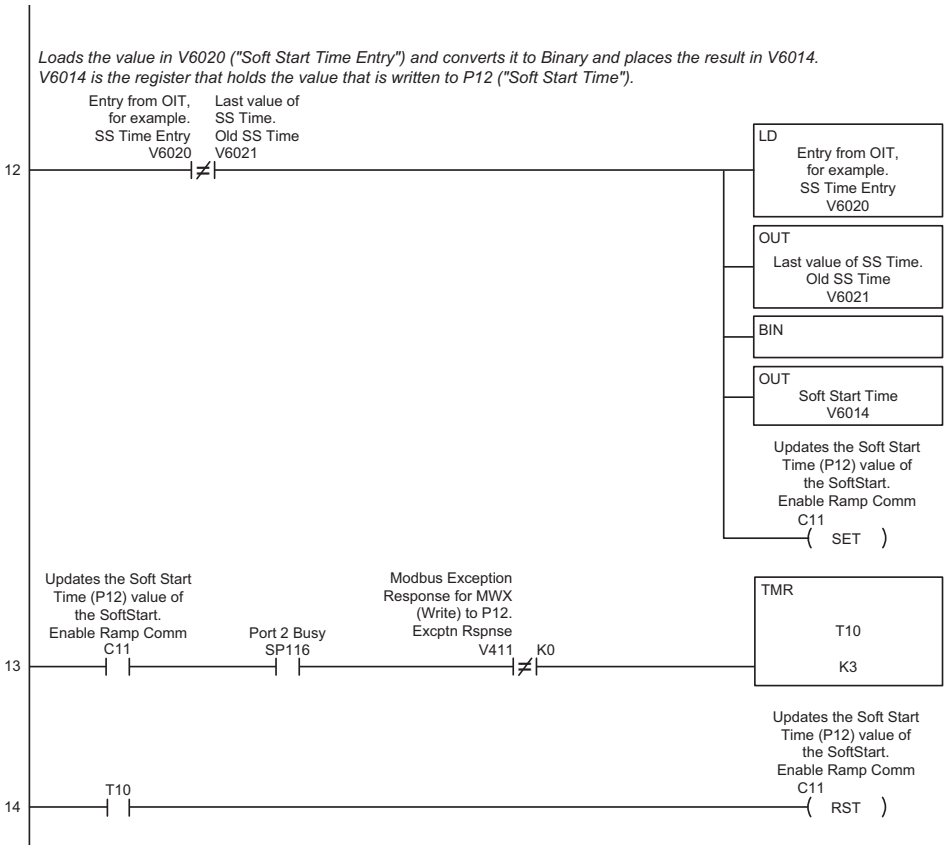
When the status of "Starting" or "Top of Ramp" turns ON, the Starting sequence is reset.



6.7.2 – *Direct*LOGIC PLC Example Program (continued)



6.7.2 – *Direct*LOGIC PLC Example Program (continued)

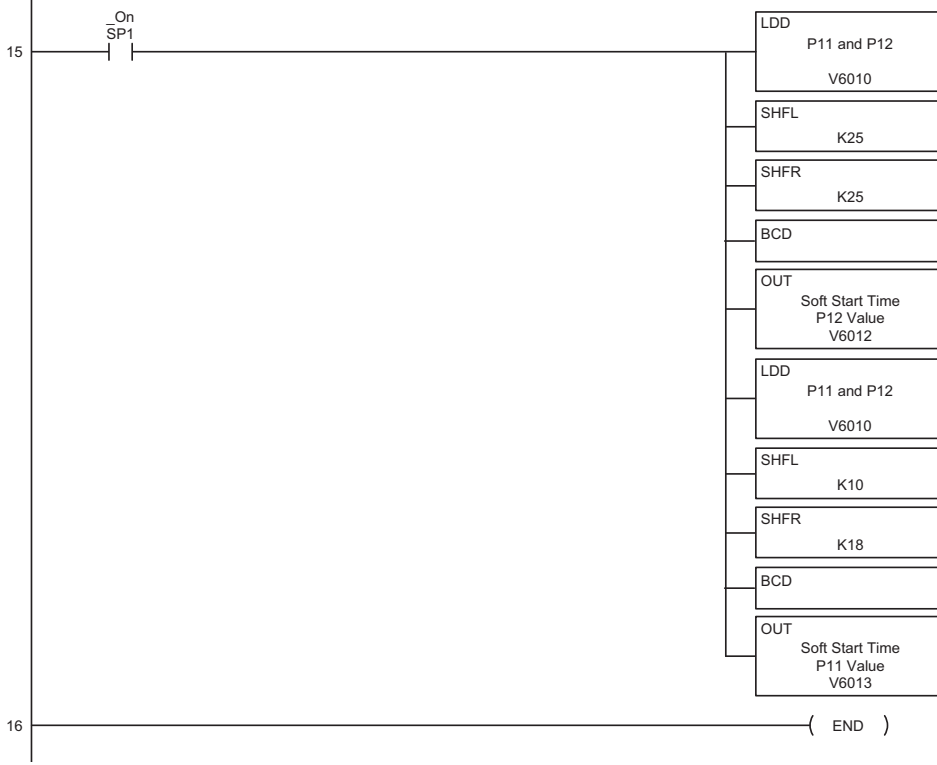


6.7.2 – *Direct*LOGIC PLC Example Program (continued)

Converts the values from P11 and P12 (which are retrieved as a single 16-bit value) into 2 separate BCD values. P11 is "Soft Start Pedestal"; P12 is "Soft Start Time". Since values are stored in bytes, a single Modbus Read will obtain 2 values.

This sequence will take the retrieved value, Shift Left by 25 bits to zero out the higher bits, then Shift Right by the same amount to put it back in its original location, convert to BCD, and this leaves the value for P12 and place it in V6012. Next, perfom a similar operation (different # of bits) to extract P11 and place the value into V6013.

These values are read from rung 10 by Exception, that is, when the user causes it.



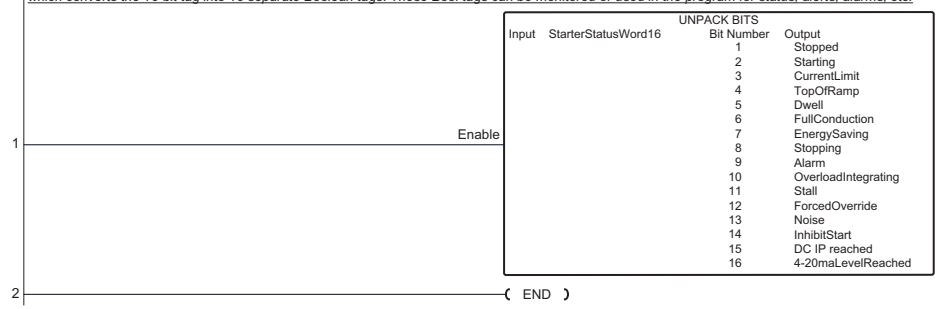
6.7.3 – Productivity3000 PLC Example Program

This information provided by AutomationDirect.com Technical Support is provided "as is" without a guarantee of any kind. These documents are provided by our Technical Support Department to assist others. We do not guarantee that the data is suitable for your application, nor do we assume any responsibility for them in your application.

This program is for illustrational purposes only, and is not intended for a true application.

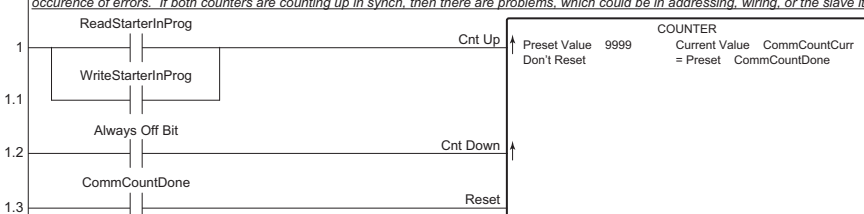
Task: StatusBits

This rung breaks down the tag "StarterStatusWord16" retrieved from the Soft Start (Parameters 8 & 9). We use the "Unpack Bits" instruction, which converts the 16-bit tag into 16 separate Boolean tags. These Bool tags can be monitored or used in the program for status, alerts, alarms, etc.

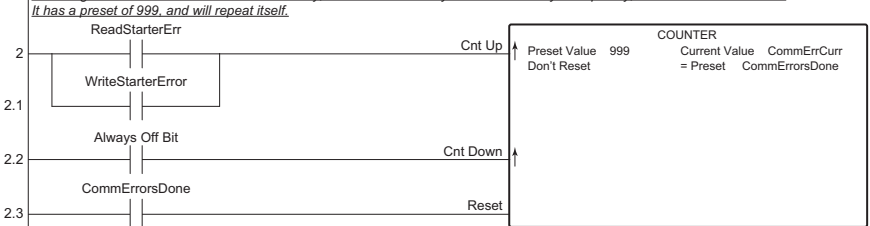


Task: Comms

This rung monitors our Comm "InProgress" attempts to the SoftStarter. The 2 inputs are the Bool tags that are assigned in the MRX and MWX instructions. It has a preset of 9999, and will reset itself. These 2 counters give us an easy way to monitor comms, to verify attempts and the occurence of errors. If both counters are counting up in synch, then there are problems, which could be in addressing, wiring, or the slave itself.



This rung monitors our Comm Errors. Ordinarily, this counter will only be activiated very infrequently, unless a slave been lost. It has a preset of 999, and will repeat itself.



6.7.3 – Productivity3000 PLC Example Program (continued)

This rung is always active and Auto-Polls the SoftStarter for Status. The tag "StarterStatusWord16" is converted into Bool tags via the Unpack Bit instruction (located in task "StatusBits"), so we will have a tag dedicated to various status conditions (Stopped, Starting, Current Limit, Alarm, etc.). These Bool tags could be used in ladder to verify operation or to set indicator lights, for example.

3 — Enable

MODBUS READ
CPU-485
Slave Node Number 1
Automatic Polling 300 0
Don't Skip Execution
Word Swap
No Mapping
Slave Modbus Starting Address 8
Modbus Decimal Addressing
Modbus Function Code 3: Read Holding Registers
Tag Name Mapping
Non-Array
Tag
StarterStatusWord16
In Progress ReadStarterInProg
Complete ReadStarterCmplt
Success ReadStarterSuccess
Error ReadStarterErr
Timeout ReadStarterO
Exception
Response String ReadStarterException

4 — StartUnitCommand — Enable
4.1 — StopUnitCommand

MODBUS WRITE
Serial Port CPU-485
Slave Node Number 1
Automatic Polling 300 20
Don't Skip Execution
No Word Swap
Map 32 bit data to 16 bit
Slave Modbus Starting Address 125
Modbus Decimal Addressing
Modbus Function Code 6: Write Single Register
Tag Name Mapping
Non-Array
Tag
P124 Starter Control
In Progress WriteStarterInProg
Complete WriteStarterComp
Success WriteStarterSuccess
Error WriteStarterError
Timeout WriteStarterTO
Exception
Response String WriteStarterException

This rung, controlled by "StarterUnit" input (usually a pushbutton input to the P3) activates the "Start Unit" sequence by setting the Bool "StartUnitCommand". Then a value of 7 ("Bus Start") is written to tag "P124 Starter Control". This will begin writing the tag value to the SoftStart until the status bits being read from the SoftStart show the unit is acting on the command. Then the sequence resets.

5 — StartUnit — StartUnitCommand (SET)
5.1 — Enable

COPY DATA
Source 7
Destination P124 Starter Control

"TopOfRamp" and "Starting" are Bool tags used to display the status of the unit. When either of these tags turns on, the sequence will reset.

6 — TopOfRamp — StartUnitCommand (RST)
6.1 — Starting
6.2 — StopUnitCommand

This rung, controlled by "StopUnit" input (usually a pushbutton input to the P3) activates the "Stop Unit" sequence by setting the Bool tag "StopUnitCommand". Then a value of 8 ("Bus Stop") is written to tag "P124 Starter Control". Again, writing will begin until status bits reflect the desired response. Then the sequence resets.

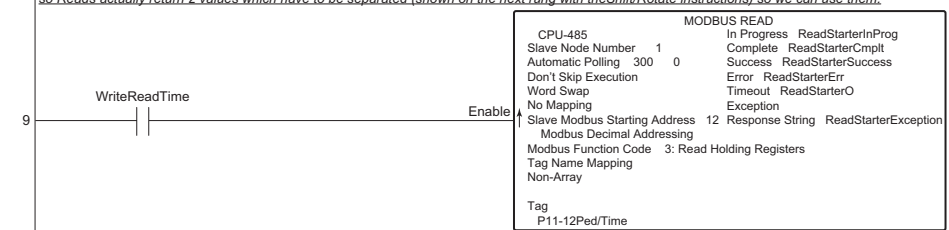
7 — StopUnit — StartUnitCommand (SET)
7.1 — Enable

COPY DATA
Source 8
Destination P124 Starter Control

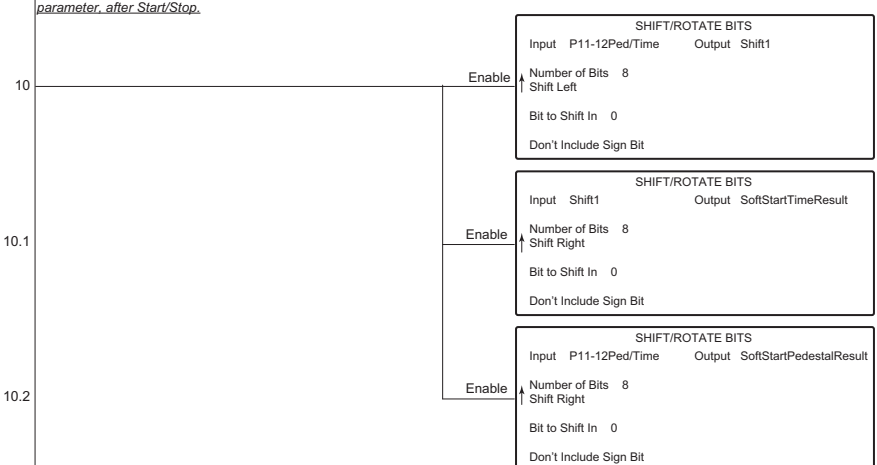
8 — Stopping — StopUnitCommand (RST)

6.7.3 – Productivity3000 PLC Example Program (continued)

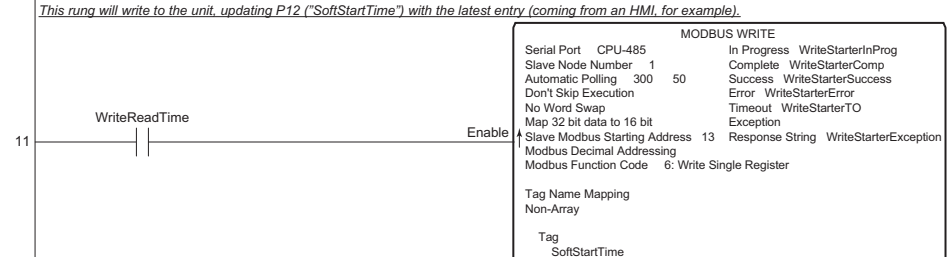
The following rungs are optional, and are included as an example of how to Read and Write to a parameter of the SoftStart. The SoftStart is 8-bit based, so Reads actually return 2 values which have to be separated (shown on the next rung with theShift/Rotate instructions) so we can use them.



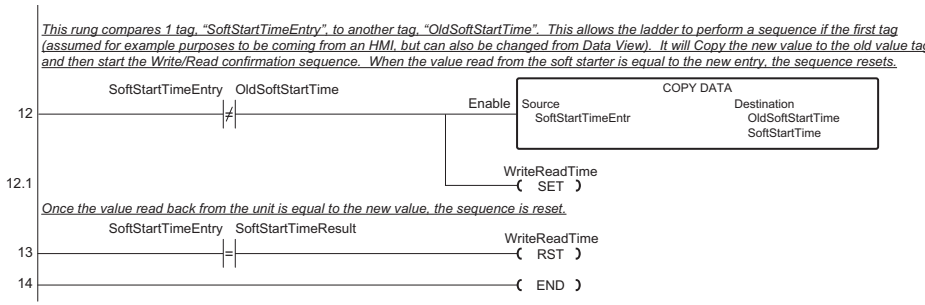
This rung takes the value in tag "P11-12", which was retrieved from the SoftStart, and performs some Shift operations on it to isolate the 2 values that actually are retrieved from the SoftStart due to its 8-bit architecture. First, the value is Shifted Left by 8 bits (to zero out the data), and the result is placed in tag "Shift1". Next, tag "Shift1" is Shifted Right by 8 bits to return the value to its original location. The resulting value is stored in tag "SoftStartTimeResult". Finally, "P11-12" is Shifted Right to move the upper byte to the lower byte and the value is stored in tag "SoftStartPedestalResult". This is the value that is used to demonstrate how to Write/Read with the unit, since the "Soft Start Time" would be the most commonly changed parameter, after Start/Stop.



This rung will write to the unit, updating P12 ("SoftStartTime") with the latest entry (coming from an HMI, for example).



6.7.3 – Productivity3000 PLC Example Program (continued)



BLANK
PAGE