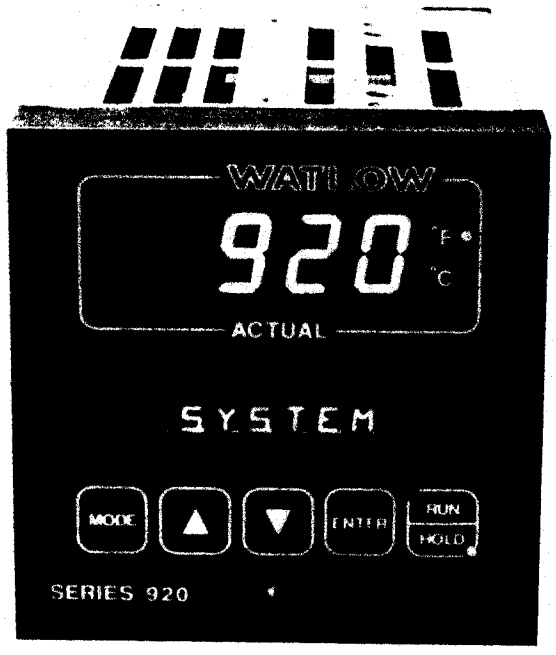
How To Use Data Communications with the Watlow Series 920



User's Manual

WATLOW

Watlow Controls, 1241 Bundy Blvd., Winona, MN 55987, Phone: 507/454-5300, Fax: 507/452-4507

W920-SA20-8944
November, 1989

$5.00

Contents

How to Use Data Communications with the Watlow Series 920

This manual is a supplement to the Series 920 User's Manual. It is for users with the data communications option; use in conjunction with the Series 920 User's Manual.

This is expert user-level material and requires previous experience with data communications.

Two Software Protocols

There are two protocols available to you. We use ANSI X3.28 Protocol, based on ANSI X3.28 - 1976 Sub categories 2.2, and A3, with the RS-422A interface to run a multiple device network. We also use XON/XOFF Protocol, a simpler protocol, to run a two device network with an RS-423A interface. XON/XOFF Protocol requires no responses to messages like the ANSI X3.28 Protocol does. XON/XOFF will also work with the RS-422A interface, but the network is limited to two devices. Likewise, ANSI X3.28 Protocol, which provides a response to every message, will work with the RS-423 interface. But again you are limited to two devices.

To select which protocol you are going to use, go into the SPCLFUNC parameters under the SETUP menu, ACCESS (5). Use the UP/DOWN keys and advance to the COM parameter. Select either XON, for the XON-XOFF communications protocol, or STX for ANSI X3.28 2.2 - A3.

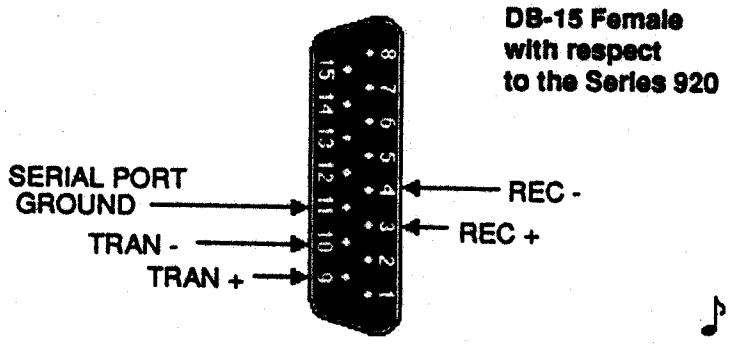
If you are using ANSI X3.28 Protocol, choose an address number for the control.

Communications Wiring

To connect your Series 920 to a computer, use the next page as a reference. Your computer hardware manual will provide the serial port pin information. Also refer to the noise prevention section in Chapter 2 of the Series 920 User's Manual. In the often noisy environments of industrial locations, it is important not to take noise isolation lightly.

RS-422A Interface Pinouts

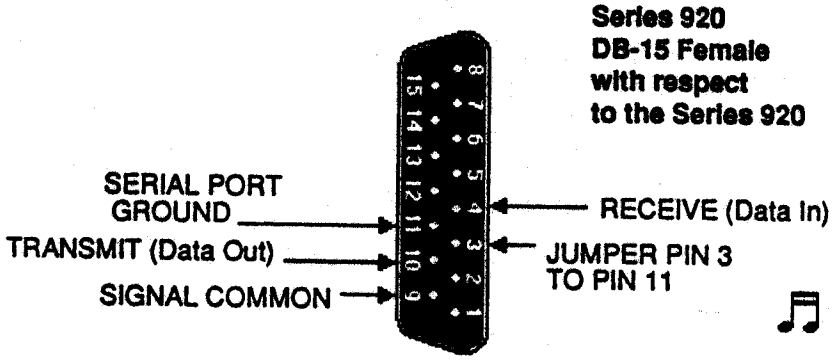
Figure 1 - RS-422 Interface, Pin Designations.



NOTE:
The Electronic Industry Association (EIA) RS-422A standard recommends a maximum 4000 ft. total network distance.

RS-423A Interface Pinouts (RS-232C Compatible)

Figure 2 - RS-423 Interface, Pin Designations.

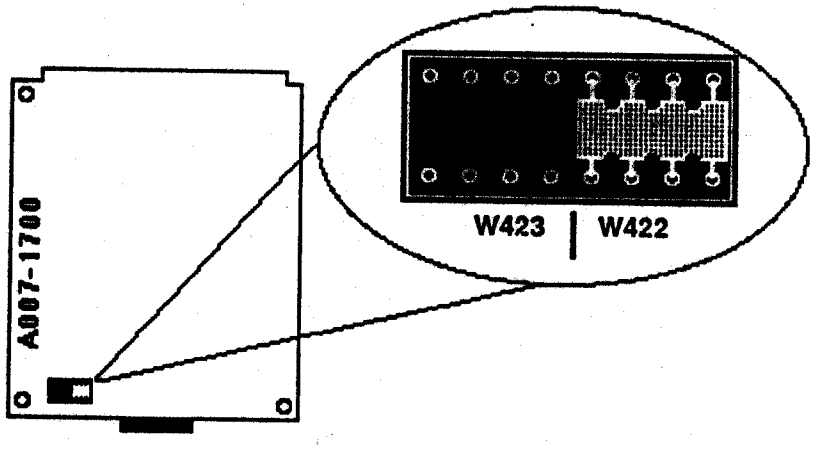


RS-422/RS-423 Jumper Selection

NOTE:
The Electronic Industry Association (EIA) RS-423A standard recommends a maximum 50 foot total point-to-point distance.

If you want to change the serial interface in your Series 920 hardware, simply move this jumper (communications board) and rewire the connectors.

Figure 3 - RS-422 / RS-423 Hardware Jumper Selection.



Network Connections

You can connect a data communication equipped Series 920 to any computer with an RS-422A or RS-423A (RS-232C compatible) serial interface. The serial interface is the key. The IBM™ PC® with an RS-232C serial output card, for instance, will talk to a single RS-423A equipped Series 920. For a multiple 920 network with the same PC, you'll need an RS-232/RS-422 converter to act as a "bus," or multiple connection point. Watlow recommends the Burr-Brown LDM 422 for that purpose. The address is: Burr-Brown, Inc., 1141 West Grant Rd., Suite 131, Tucson, AZ 85705, Phone: (602) 624-2434, Fax: (602) 623-8965.

Connecting the Control and the Computer

Remove power from both the Series 920 and your computer before connecting them together. This prevents noise or static interference entering the data communication lines.

As soon as you connect the data communications line(s), you're ready to apply power to your system.

Communications Software

Watlow has available a communications program for the Series 920 and the IBM-PC® or MS-DOS™ compatible. The Watlow programs are written in Microsoft™ Quick BASIC 4.5, and may be obtained from your Watlow field sales representative.

If you want to write your own communications software for the Series 920, or edit the program, you will be able to do that with the information provided in this manual.

Series 920 Communication Parameters

To communicate with the Series 920, set your computer for these parameters:

Baud Rate	=	1200
Parity	=	Odd
Data Bits	=	7
Start Bits	=	1
Stop Bits	=	1

All the parameters listed above are factory selected.

ASCII and Series 920 Information

The next page shows you ASCII and control key equivalent tables. Following that, you will find a detailed explanation of the Series 920 syntax and command structure for each of the two protocols.

ASCII Character Set

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
00	00	NUL	16	10	DLE	32	20	SP	48	30	0
01	01	SOH	17	11	DC1	33	21	!	49	31	1
02	02	STX	18	12	DC2	34	22	"	50	32	2
03	03	ETX	19	13	DC3	35	23	#	51	33	3
04	04	EOT	20	14	DC4	36	24	$	52	34	4
05	05	ENQ	21	15	NAK	37	25	%	53	35	5
06	06	ACK	22	16	SYN	38	26	&	54	36	6
07	07	BEL	23	17	ETB	39	27	'	55	37	7
08	08	BS	24	18	CAN	40	28	(	56	38	8
09	09	HT	25	19	EM	41	29	)	57	39	9
10	0A	LF	26	1A	SUB	42	2A	*	58	3A	:
11	0B	VT	27	1B	ESC	43	2B	+	59	3B	;
12	0C	FF	28	1C	FS	44	2C	,	60	3C	<
13	0D	CR	29	1D	GS	45	2D	-	61	3D	=
14	0E	SO	30	1E	RS	46	2E	.	62	3E	>
15	0F	SI	31	1F	US	47	2F	/	63	3F	?

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
64	40	@	80	50	P	96	60	`	112	70	p
65	41	A	81	51	Q	97	61	a	113	71	q
66	42	B	82	52	R	98	62	b	114	72	r
67	43	C	83	53	S	99	63	c	115	73	s
68	44	D	84	54	T	100	64	d	116	74	t
69	45	E	85	55	U	101	65	e	117	75	u
70	46	F	86	56	V	102	66	f	118	76	v
71	47	G	87	57	W	103	67	g	119	77	w
72	48	H	88	58	X	104	68	h	120	78	x
73	49	I	89	59	Y	105	69	i	121	79	y
74	4A	J	90	5A	Z	106	6A	j	122	7A	z
75	4B	K	91	5B	[	107	6B	k	123	7B	{
76	4C	L	92	5C	\	108	6C	l	124	7C	|
77	4D	M	93	5D	]	109	6D	m	125	7D	}
78	4E	N	94	5E	^	110	6E	n	126	7E	~
79	4F	O	95	5F	_	111	6F	o	127	7F	DEL

Table 1 - ASCII Character Set.

ASCII Control Characters (Partial Set)

ASCII Char.	Ctrl Key Equiv.	Definition	Dec. Equiv.	Hex. Equiv.
ENQ	Ctrl E	Enquiry	5	05
ACK	Ctrl F	Acknowledge	6	06
NAK	Ctrl U	Neg. Acknowledge	21	15
STX	Ctrl B	Start of Text	2	02
ETX	Ctrl C	End of Text	3	03
EOT	Ctrl D	End of Transmission	4	04
DLE	Ctrl P	Data Link Escape	16	10
CR	Ctrl M	Carriage Return	13	0D
DC1	Ctrl Q	XON	17	11
DC3	Ctrl S	XOFF	19	13

Table 2 - ASCII Control Characters. (Partial Set)

Series 920 General Message Syntax

As soon as you link the devices, you'll be able to talk to the Series 920 using ASCII characters.

The Series 920 will respond to any parameter showing in the alphanumeric display, plus some others. The control will respond to either upper or lower case ASCII characters from your computer keyboard.

Both protocol/interface combinations will respond to the general syntax, providing the commands or queries are correctly transmitted. However, the ANSI X3.28 Protocol requires beginning and ending characters, and the XON/XOFF Protocol requires ending characters. We'll look at those shortly.

Message Syntax

Messages from your computer to the Series 920 must take this general form. All commands do not require the full number of data fields.

Command <Space> Data.1 <Space> Data.2 <Space> Data.3... Data.N

"Command" is a character set to which the Series 920 will respond. The brackets "< >" enclose a non-literal description. "Space" is simply a delimiter, an ASCII space character (Hex 20). "Data Fields" are parameters and values specific to a command; the number of possible data fields depends on the particular command you use. Data 1 is here abbreviated, "Data.1", Data 2 is "Data.2" and so on.

In the syntax explanations ahead, we'll show you the specific arguments for each command. It will speed the process, though, if you remember this general syntax.

Data Rules

Data fields are parameters and values specific to particular commands. These rules govern their use. Specific data for each command is listed later in this chapter.

- Data will be ASCII 0 through 9, unless otherwise noted.
- Data can go up to four total characters, including a minus sign.
- Data can use leading zeros. (Up to 4 digits.)
- Data does not use decimal points, therefore, a number must have units equal to the least significance of the data value. If, for example, a parameter's value is 1, and the least significant digit of the parameter is 0.1, the number sent as the data would be 10. That is to say the value is interpreted as 10 tenths.
- Data returned from the unit that has a value of "*" (asterisk), means that the value is unprogrammed.

Command List

These commands, represented by their respective ASCII characters, will enable you to program the Series 920 from your computer keyboard. More detailed descriptions of the commands are on the pages noted.

?	Finds the value of a specific parameter.	p. 15
=	Sets a specific parameter to a specific value.	p. 15

Example Format

For your benefit, we're presenting message/response examples with syntax required for Series 920 communication. Information bracketed by < > indicates a description, rather than literal characters. We show each ASCII character that you must transmit to the Series 920, including space between the characters. (A "space" is itself an ASCII character, hex 20). For clarity, we also represent each ASCII character as a hexadecimal pair. The pairs are spread apart on the page for easy reading. However, electronic devices "see" the hex pairs all together in "strings," with no spaces in between.

For instance, from the example just below, you want to set the Alarm 1 Low (A1L) parameter to 500°. Notice the syntax just below which uses the "=" command.

= <Space> A1L <Space> 500

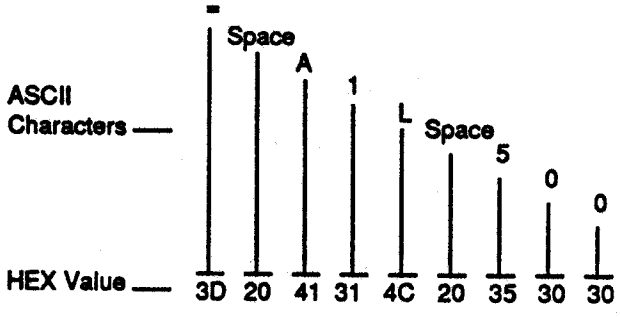


Figure 4 - Series 920 General Message Syntax Example.

To send the message on Page 8, Figure 4, you key the ASCII characters into your computer, or write them into your program. The computer, in turn, will send a string similar to the one at the bottom of the example, 3D2041314C20353030.

Notice that we haven't mentioned protocol here, or any characters added to this syntax by a protocol. With XON/XOFF, the message above can be transmitted with only an additional Carriage Return <CR> (hex 0D) character at the end. However, the ANSI X3.28 Protocol requires an envelope of Start of Text <STX> (hex 02) and End of Text <ETX> (hex 03) characters around the information you see above. You'll learn how to do that in the pages ahead.

XON/XOFF Protocol for RS-423A

XON/XOFF (flow control) Protocol allows a communicating device (either a 920 or the host) to suspend transmission of all messages from the other device, and then to continue transmission when it's again ready.

The device that needs to suspend transmission sends the XOFF character (hex 13) to stop the other device's transmitter, and XON (hex 11) to restart it. Note that technically any character will restart the transmitter, but only the XON character is not a part of any regular message that may be transferring.

Messages transmit according to the syntax described in the XON/XOFF formats which follow for each command.

The XON/XOFF Protocol requires a Carriage Return character (hex 0D) at the end of every message.

How To Start and Stop Communicating with the Series 920 and XON/XOFF

Starting communication with **XON/XOFF Protocol** is simple. You just configure your computer to agree with the Series 920 communication parameters and open its serial communication port in software. Then begin to "talk" by transmitting a message to the Series 920. You stop communicating with XON/XOFF Protocol simply by ceasing to send messages.

XON/XOFF "=" Command Example

The general command syntax is the one you've already seen. Each command uses a slightly different variation of it, depending on the number of arguments required for a message. The control must be in the HOLD mode to accept a change.

• You want to change the Alarm 1 Low (A1L) value to 500°. The "=" Command will do the job.

The syntax with XON/XOFF Protocol requires an ending Carriage Return <CR>.

"=" Command Syntax with XON/XOFF Protocol:
= <space> Data.1 <space> Data.2 <CR>

With the "=" Command, Data.1 is the Series 920 parameter, in this case Alarm 1 Low, A1L. Data.2 is the value you want to set for that parameter, in this example, 500.

Enter in ASCII:
= <space> A1L <space> 500 <CR>
The hex string will be:
3D2041314C203530300D

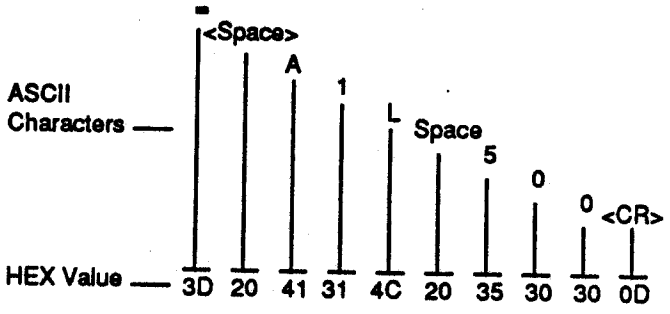


Figure 5 - XON/XOFF "=" Command Example.

Response from the Series 920:
Sends a "XOFF" when a carriage return is received and then an "XON" when the unit is done processing the command.

• The complete list of "=" Command data (parameters and value limits) is in Table 6, Pages 20 - 21.

XON/XOFF "?" Command Example

You want to know the Alarm 1 Low (A1L) value. The "?" uses a variation of the message syntax shown just below. **This protocol requires an ending carriage return character.**

"?" Command syntax with XON/XOFF Protocol:
? <space> Data.1 <CR>

Enter in ASCII:
? <space> A1L <CR>
The hex string will be:
3F2041314C0D

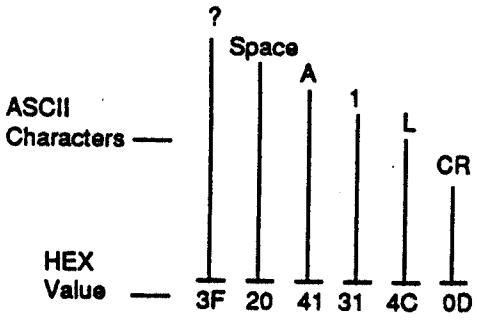
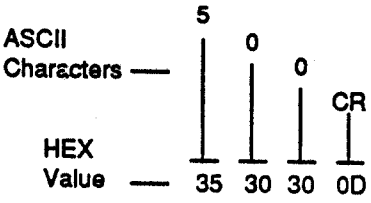


Figure 6 - XON/XOFF "?" Command Example.

The value of A1L will be between RAL (Range Low) and RAH (Range High), say, 500.

Response from the Series 920:
<XOFF> <XON> <current value of A1L> <CR>

The hex response string is:
3530300D



ANSI X3.28 Protocol

The ANSI X3.28 Protocol provides high quality communications by requiring a response to every message. With a multiple device or "multidrop" network, this protocol prevents confusion among the separate devices. Furthermore, if noise occurs somewhere in the system, no parameter will change because noise can't comply with the protocol.

By placing messages inside a protocol envelope, the messages are protected. In the examples to come you'll see how this works.

The ANSI X3.28 Protocol requires STX characters at the beginning of a message and ETX characters at the end.

Device Address

If you are using the ANSI X3.28 Protocol, you must have a device address (identification) number. A Watlow RS-422A multidrop network can handle up to 10 devices with this protocol. Set the address number with the Series 920 SETUP - SPCLFUNC menu (ACCESS 5), at the COM ID parameter.

Starting Communications in ANSI X3.28 Protocol

Here's the syntax for starting communications with ANSI X3.28 Protocol. The master device, your computer, must initiate the data link. The example below uses the ASCII number 4 as a Series 920 device address.

Enter in ASCII, using this syntax: <Address # 4><ENQ>

ASCII Characters	4	<ENQ>
HEX Value	34	05

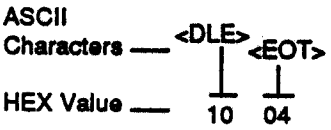
Response from the 920:
<Address # 4><Acknowledge (ACK)>

ASCII Characters	4	<ACK>
HEX Value	34	06

Stopping Communications in ANSI X3.28 Protocol

The master device, your computer, must end communications with Device #4 by using Data Link Escape (DLE) and End of Transmission (EOT) characters.

Enter in ASCII: <DLE><EOT>



Response from the 920:
None

ANSI X3.28 "=" Command Example

The "=" Command sets a specific 920 parameter to a specific value. The control must be in the HOLD mode to accept the change. The general command syntax applies to all commands. The definition and number of arguments depends on the command itself. See Table 6, Pages 20 - 21.

In this example, you want to change the Alarm 1 Low value to 500°. Here, the "=" Command will do the job.

'"=" Command Syntax with ANSI X3.28 Protocol:
<STX> = <space> Data.1 <space> Data.2 <ETX>

With the "=" Command, Data.1 is the Series 920 parameter, in this case Alarm 1 Low , A1L. Data.2 is the value you want to set for that parameter, in this example, 500.

Enter in ASCII:
<STX> = <space> A1L <space> 500 <optional carriage return> <ETX>
The hex string is:
023D2041314C2035303003

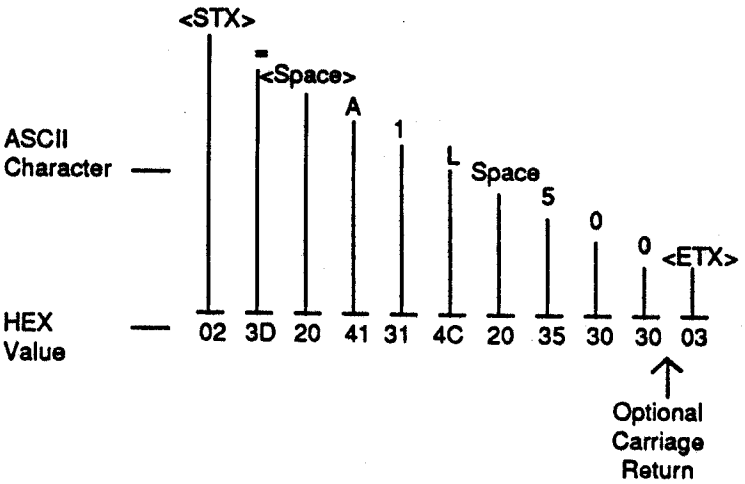


Figure 7 - ANSI X3.28 "=" Command Example.

Response from the Series 920:
<ACK>
The hex response string is:
06

• You'll find the the complete list of "=" Command arguments (parameters and value limits) in Table 6, Pages 20 - 21.

ANSI X3.28 "?" Command Example

You need to know the Alarm 1 Low value (A1L). The "?" uses a variation of the message syntax shown just below. **This syntax requires the protocol start of text and end of text characters.**

"?" Command syntax with ANSI X3.28 Protocol:
<STX> ?<space> <Data.1> <ETX>

Enter in ASCII:
<STX> ? <space> <A1L> <optional carriage return> <ETX>
The hex string will be:
023F2041314C03

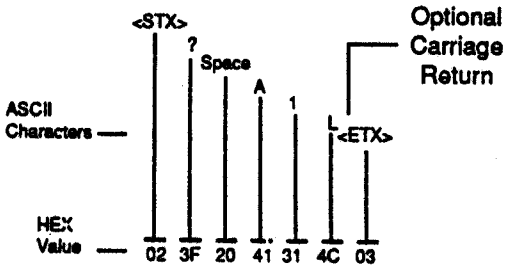
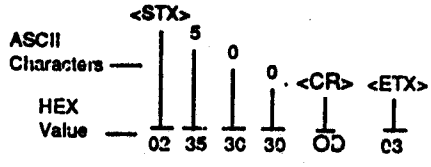


Figure 8 -
ANSI X3.28 "?"
Command Example.

First response from the Series 920:
<ACK>
The <ACK> hex response string is:
06

Your computer's confirming response:
<EOT>
The <EOT> response hex string is:
04

Second response from the Series 920:
<STX> <current A1L value> <carriage return> <ETX>
The hex string is:
023530300D03



Your computer's next response:
<ACK> or < NAK> (if the message needs to be repeated).
The hex string is:
06 or 15

Final response from the Series 920:
<EOT>
The hex string is:
04

"?" Command

The "?" Command finds the specific value of a Series 920 parameter (Data.1). Tables 3 and 6 provide the complete list of parameters you may use, plus responses. You may use the "?" Command in either the RUN or HOLD modes. Data returned from the unit, with a value of "*" (asterisk), are unprogrammed parameters.

"=" Command ♪

"The "=" Command sets a specific Series 920 parameter (Data.1) to a specific value (Data.2) when the unit is in the HOLD mode. Use Table 6 to select parameters (Data.1) in the lefthand column, with low and high limit or code values (Data.2) in the three center columns. Use the table with either syntax.

♪

NOTE:
You may write to a Series 920 parameter with the "=" Command only when the unit is in the HOLD mode.

"?" Command Syntax

The following commands in Table 3 on Page 16, are "?" commands only. The syntax for these "?" commands, not including protocol characters, would be:

? <Space> Data.1.

The response from the "?" commands, not including protocol characters, is in the "Respns" column.

Data.1	Respns	Information	Comments
ACT	ACTUAL	Actual process value	Between RAL and RAH
AFL	See Page 17.	Finds all programmed files.	
ALM	0	No alarms occurring	"= ALM 0 " will clear alarms.
	1	A1H occurring	
	2	A1L occurring	
	4	A2H occurring	
	8	A2L occurring	
DIP	0	All DIP Switches ON	
	1	DIP Switch 1 OFF	
	3	DIP Switch 1-2 OFF	
	7	DIP Switch 1-3 OFF	
	F	DIP Switch 1-4 OFF	
	1F	DIP Switch 1-5 OFF	
	3F	DIP Switch 1-6 OFF	
	7F	DIP Switch 1-7 OFF	
	FF	All DIP Switches OFF	
ER1	0	No error	Cleared when ER1 is read
	1	Internal RAM error	
	2	External RAM error	
	3	Bad battery error	
	5	A/D conversion/open input error	
	9	High reference out of limit	
	10	Cold junction offset out of limit	
	11	RTD gain out of limit	
	12	RTD zero out of limit	
	13	0-5V/4-20mA out of limit	
	14	Interpolation/overrange	
	15	Ambient temperature overrange	
	16	0-5V/4-20mA offset out of limit	
	20	F/V offset out of limit	
	21	F/V gain out of limit	
	22	Stack overflow error	
ER2	0	No error	Cleared when ER2 is read
	1	Transmit buffer overflow	
	2	Receive buffer overflow	
	3	Framing/overrun error	
	5	Parity error	
	6	Talking out of turn	
	7	Invalid reply error	
	20	Command not found	
	21	Parameter not found	
	22	Incomplete command line	
	23	Invalid character	
	24	Number of chars. overflow	
	25	Input out of limit	
	26	Read only command	
	28	Write only command	
	30	Request to RUN invalid	
	31	Request to HOLD invalid	
	32	Command invalid in RUN mode	
	33	Self test mode not active	
	34	Memory write not successful	
	35	Number of steps stored is > 99	
	36	No file found	
	37	No step found	
	38	No asterisk input allowed	
	39	Infinite loop error	
	40	File change error	
FST	See Page 17.	Finds number of steps in a file.	
JREM	0 - 100	The # of jumploops remaining.	
KEY	0	None	Cleared when key is read
	1	Mode	
	2	Enter	
	4	Incrememt	
	8	Decrement	
	10	Run/Hold	
MDL	See Page 17.	Model # and Revision #.	
MTR	See Page 18.	Monitors the current step.	
RUN	0	HOLD mode	
	1	RUN mode	
STP	See Page 19.	Finds data programmed in step.	

Table 3 - "?" Command Data and Responses, Partial Set.

"? AFL" Command

This command asks the Series 920 for the file numbers of files that are programmed.

The syntax for the command (not including protocol characters) is:
?<Space> AFL

The command will return the following data in ASCII numbers:

Data.1 <Space> Data.2 <Space> Data.3 <Space> ... Data.n

Data.1 will always be 1, i.e., File #1. File #1 will always appear, even with a cold-started unit when that file is defaulted to a one step STOP step program. Data.2 through Data.n are the file numbers of the other programmed files.

"? FST" Command

This command asks the number of steps programmed in the specified file.

The syntax for this command (not including protocol characters) is:
? <Space> FST <Space> <File #>

This command will return the number of programmed steps in the specified file in a one or two character ASCII number.

"? MDL" Command

This command asks the 920 for the type of unit and the level of software. The model number will appear as 1 character. The syntax of the function is as follows:
? <Space> MDL

This function will return the following information:

920<Model number> <Space> <Software revision level>

"? MTR" Command

The ? MTR command monitors the current RUNning step, or the step that will execute from the HOLD mode if RESUME is entered. Data returned from the 920 that has a value of "*" (asterisk) means the value is unprogrammed.

The syntax for this command is:
?<Space> MTR

There are no arguments to the command. The command will return the following data:

<File #> <Space> <Current Step #> <Space> <Current Step Type> <Space> Data.4 <Space> Data.5 ... Data.n

Table 4 below shows how to interpret the response syntax. Note that each argument is an ASCII decimal number representing a specific value for each parameter listed in the table. A response would be:

<3> <space> <11> <space> <1> <space> <375> <space> <2> <space> <30> <space> <0> <space> <1> <space> <1>

This tells you that the current step is File#3, Step#11 a SETPOINT step, the SP is 375°. Step Time is 2:30:00, EV1 and EV2 are ON.

Table 4 uses the following abbreviations:

WPV = ACTUAL waitfor temperature
WHR = Wait for hours programmed
WMN = Wait for minutes programmed
WRH = Wait for remaining hours
WRM = Wait for remaining minutes
WRS = Wait for remaining seconds
ADA = Autostart days accumulated
RH = Real time hours
RM = Real time minutes

Table 4 - "? MTR" Command Response Data. <> = Non-literal Description.

NOTE:

You will see only one of the "SETPOINT" data fields depending on how your control is programmed.

	PRG = Rate	PRG = Time	Step Type				
	SETPOINT	SETPOINT	JUMPLOOP	WAITFOR	AUTOSTART	STOP	LINK
File #	<File#>	<File#>	<File#>	<File#>	<File#>	<File#>	<File#>
Step #	<Step#>	<Step#>	<Step#>	<Step#>	<Step#>	<Step#>	<Step#>
Type Code	1	1	2	3	4	5	6
Data.4	<SP>	<SP>	<JS>	<WPV>*	<DAY>*		<Link File#>
Data.5	<RATE>	<HOUR>	<JC>	<WHR>	<HOUR>*		
Data.6	<EV1>	<MIN>		<WMN>	<MIN>*		
Data.7	<EV2>	<SEC>		<WRH>	<ADA>		
Data.8		<EV1>		<WRM>	<RH>		
Data.9		<EV2>		<WRS>	<RM>		

"? STP" Command

The "? STP" command reads a given step in the Series 920 program space (99 steps total). Data returned from the unit, that have a value of "*" (asterisk) mean that the value is unprogrammed.

The syntax for this command is:
?<Space> STP <Space> <File #> <Space> <Step #>

The command will return the following information:
<Step Type> <Space> Data.2 <Space> Data.3 ... Data.n

Table 5 below shows how to interpret the response syntax. Note that each argument is an ASCII decimal number representing a specific value for each parameter listed in the table.

A response for a Step# 8, in File#4 would be:
<1> <space> <235> <space> <1> <space> <20> <space> <15> <space> <1> <space> <0>

This tells you that Step# 8 in File#4 is a SETPOINT step, SP is 235°. Step Time is 1:20:15, EV1 is ON and EV2 is OFF.

In Table 5, the abbreviation "WPV" is "Wait for ACTUAL temperature."

Table 5 - "? STP" Command Response Data. <> = Non-literal Description.

♪ **NOTE: You will see only one of the "SETPOINT" data fields depending on how your control is programmed.**

	PRG = Rate	PRG = Time	Step Type				
	SETPOINT	SETPOINT	JUMPLOOP	WAITFOR	AUTOSTART	STOP	LINK
Type Code	1	1	2	3	4	5	6
Data.2	<SP>*	<SP>*	<JS>	<WPV>*	<DAY>*		<Link File#>
Data.3	<RATE>*	<HOUR>*	<JC>	<WHR>*	<HOUR>*		
Data.4	<EV1>*	<MIN>*		<WMN>*	<MIN>*		
Data.5	<EV2>*	<SEC>*					
Data.6		<EV1>*					
Data.7		<EV2>*					

"?" & "="

These commands are "?" and "=" commands. The syntax for the "?" commands, not including protocol characters, is:
? <space> Data.1
The response from the "?" commands, not including protocol characters, is per the Data.2 column.

The syntax for the "=" commands, not including protocol characters, is:
= <space> Data.1 <space> Data.2

Table 6 - "=" Command and "?" Command Data.

Data.1	Data.2			
	Low Limit	High Limit	Code	Function
A1H*	RAL value	RAH value		Set Alarm 1 High value
A1L*	RAL value	RAH value		Set Alarm 1 Low value
A2H*	RAL value	RAH value		Set Alarm 2 High value
A2L*	RAL value	RAH value		Set Alarm 2 Low value
ALM			0	Clears alarms
AT1			0	Set Alarm 1 to Process
			1	Set Alarm 1 to Deviation
AT2			0	Set Alarm 2 to Process
			1	Set Alarm 2 to Deviation
AUX1			0	AUX1 - Alarm Output
			1	AUX1 - Event Output
AUX2			0	AUX2 - Alarm Output
			1	AUX2 - Event Output
CAL*	-90°F -50°C -50 Units	90°F 50°C 50 Units		Set Calibration offset
CFU	0	2	0	Display °F
			1	Display °C
			2	Display Units (PVUs)
CTC	1	60		Set Cycle Time-Cool
CTH	1	60		Set Cycle Time-Heat
DB*	-36°F -20°C -20 Units	36°F 20°C 20 Units		Set Dead Band
EV1	0	1	0	Set EV1 OFF
			1	Set EV1 ON
EV2	0	1	0	Set EV2 OFF
			1	Set EV2 ON
GS*	0°	18°F 10°C 10 Units		Set Guaranteed Soak
HPWR	0	100		Set % High Power
IN			0	Type J
			1	Type K
			2	Type T
			3	Type R
			4	Type S
			5	Type B
			6	4-20mA
			7	RTD wholes
			8	RTD tenths
			9	0-5 volts
LAT	0	1	0	Set to non-latched alarm
			1	Set to a latched alarm
LOCK	0	2		Lock front panel
LPWR	0	100		Set % Low Power

*** When the 920 RTD input is 0.1°, these parameters will have an implied decimal point to the left of the least significant digit.**

Data.1	Data.2			
	Low Limit	High Limit	Code	Function
OUT			0 1 2 3	Heat Only Cool Only Heat/Cool Cool/Heat
PBC*	0	900°F 500°C 500 Units		Set Prop. Band-Cool
PBH*	0	900°F 500°C 500 Units		Set Prop. Band-Heat
PRG			0 1	Set to Time Duration Set to Ramp Rate
RAH*	Min. IN Range	Max. IN Range		Set Range High
RAL*	Min. IN Range	Max. IN Range		Set Range Low
RBC	0	7		Set Rate Band-Cool
RBH	0	7		Set Rate Band-Heat
RSC	0	500		Set Reset-Cool
RSH	0	500		Set Reset- Heat
RTC	0	500		Set Rate-Cool
RTD			0 1	JIS Curve DIN Curve
RTH	0	500		Set Rate-Heat
SP*	RAL	RAH		Enter fixed Set Point
TI				See Page 23

Table 6 - Continued

*** When the 920 RTD input is 0.1°, these parameters will have an implied decimal point to the left of the least significant digit.**

"= CLRF" Command

The "= CLRF" command clears a given file# from 1 to 10 in the Series 920 program space (99 steps total). This command can only be performed in the HOLD mode.

The syntax of the command (not including protocol characters) is:
= <Space> CLRF <Space> <File #>

This command will not return an error message even though the file number may not exist.

"= HOLD" Command

This command causes a 920 program to HOLD. If the program is already HOLDing, an error will occur.

The syntax for this command (not including protocol characters) is:
= <Space> HOLD <Space> 1

Check the HOLD status with "? RUN".

"= RSUM" Command

This command causes a Series 920 program to RESUME. If the program is already RUNning, an error will occur.

The syntax for this command (not including protocol characters) is:
= <Space> RSUM <Space> 1

Check the RUN status with "? RUN".

"= STRT" Command

This command causes a 920 program to start. If the program is already RUNning, an error will occur.

The syntax for this command (not including protocol characters) is:
= <Space> STRT <Space> <File #> <Space> <Step#>

Check the RUN/HOLD status with the "?RUN" command.

"= STP" Command

NOTE:
You may write to a Series 920 step with the "STP" Command only when the unit is in the HOLD mode.

The "= STP" command is used to program a given step in a file. Remember that total program space is 99 steps. You can use the "=STP" command only in the HOLD mode.

The syntax for this command (not including protocol characters) is:
= <Space> STP <space><File#><space><Step#><space><Step Type Code><space><Data.5><space><Data.6><space>...<Data.ON>

Table 7 below shows how to interpret the syntax. Note that each Data field is an ASCII decimal number representing a specific value for each parameter listed in the table. The abbreviation "WPV" is "Wait for ACTUAL temperature."

Table 7 - "=STP" Command Data. <> = Non-literal Description.

A syntax example for a SETPOINT Step# 25, in File#7, SP at 255° for 1 hour, 36 minutes, 58 seconds with EV1 ON and EV2 OFF would be:

= <space> <STP> <space> <7> <space> <25> <space> <1> <space> <255> <space> <1> <space> <36> <space> <58> <space> <1> <space> <0>

NOTE:
You will see only one of the "SETPOINT" data fields depending on how your control is programmed.

	PRG = Rate	PRG = Time	Step Type				
	SETPOINT	SETPOINT	JUMPLOOP	WAITFOR	AUTOSTART	STOP	LINK
File #	<File#>	<File#>	<File#>	<File#>	<File#>	<File#>	<File#>
Step #	<Step#>	<Step#>	<Step#>	<Step#>	<Step#>	<Step#>	<Step#>
Type Code	1	1	2	3	4	5	6
Data.5	<SP>*	<SP>*	<JS>	<WPV>*	<DAY>*		<Link File#>
Data.6	<RATE>	<HOUR>*	<JC>	<WHR>*	<HOUR>*		
Data.7		<MIN>*		<WMN>*	<MIN>*		
Data.8		<SEC>*					
Data.9		<EV1>*					
Data.10		<EV2>*					

* These commands can accept an asterisk to indicate that they are not programmed.

"= TI" Command

The "= TI" Command is used to set the Series 920 real time clock in hours, minutes, and seconds.

The syntax for this command (not including protocol characters) is:
= <Space>TI <space> <Data.1> <space> <Data.2> <space> <Data.3>

The arguments, in ASCII numbers, must always be:
Data.1 = Real time hours, range, 0 to 23.
Data.2 = Real time minutes, range, 0 to 59.
Data.3 = Real time seconds, range, 0 to 59.

The "= TI" Command can only be used when the unit is in the HOLD mode.

NOTE:
You may write to the Series 920 real time clock with the "=TI" Command only when the unit is in the HOLD mode.

"? TI" Command

The "? TI" command asks Series 920 real time clock for its hours, minutes, and seconds.

The syntax for this command (not including protocol characters) is:
?<Space> TI

The command will return the following data :

<Data.1> <space> <Data.2> <space> <Data.3>

The arguments, in ASCII numbers, will always be:
Data.1 = Real time hours, range, 0 to 23.
Data.2 = Real time minutes, range, 0 to 59.
Data.3 = Real time seconds, range, 0 to 59.

Watlow Controls

Watlow Controls is a division of Watlow Electric Manufacturing Company of St. Louis, Missouri. Watlow is an established manufacturer of industrial electric heating products, in business since 1922. Watlow boasts the ability to begin with a full set of specifications and to complete an industrial product that is manufactured totally in-house, in the U.S.A. Products designed and manufactured by Watlow are electric heating elements, sensors, electronic temperature controls and power switching devices.

The Winona operation has been designing solid state electronic control devices since 1962, and has earned the reputation as an excellent supplier to original equipment manufacturers. These OEMs depend upon Watlow Winona to provide compatibly engineered controls which they can incorporate into their products with confidence.

Watlow Controls resides in a 100,000 square foot marketing, engineering and manufacturing facility in Winona, Minnesota.