TechNote

ASCII Message Transmitting with CTC Controllers

30

Control Technology Corporation, Hopkinton, MA • 800.282.5008 • www.ctc-control.com

Current versions of the models 2600 and 2700 controllers allow you to create “messages” to be transmitted from the controller’s RS-232 ports under the control of your program. This is useful for a variety of purposes, including:

- Data logging (by connecting a serial printer to the controller).
- Fault logging (by connecting a serial printer to the controller).
- Transmitting numeric data from one controller to another.
- Out-bound transmission of a “service request” to a computer.
- Interface to serial display devices.
- Interface to “foreign” (non-CTC) devices requiring ASCII commands.

In the 2800 and 2400 Series controllers, message transmission may be accomplished over either Channel A or Channel B and will take place at the baud rate of the selected channel (fixed or 9600). Channels maybe alternated. The baud rate for Channel B is fixed at 9600 baud; the baud rate for Channel A is selectable. For information on setting the baud rate, see your controller’s installation guide.

What Is ASCII?

ASCII stands for American Standard Code for Information Interchange, and is a code by which each letter of the alphabet, each number from 0 to 9. and certain control codes are each represented by a number from 0 to 127. This code allows all of the information that might exist on a printed page to be transmitted as a series of numbers, each number representing one character. This means that multidigit numbers are actually represented in ASCII as a series of numbers: one number for each character, or digit, of the multidigit number being represented.

Message Storage

The source of the messages is the controller’s Data Table (unfortunately, this limits the Data Table’s use for program parameters). ASCII messages are loaded into the rows of the Data Table (these may be configured for up to 32 columns, or “characters,” each), following the rules defined below. The controller can transmit any given row by sending the desired row number to a special purpose register, register 12001 (e.g.: -STORE 1 TO REG12001). To send messages longer than 32 characters, store them in multiple rows of the Data Table, and then send the rows in successive order. Because the carriage return and line feed commands must be sent as explicit ASCII characters, such a long message may be made to appear on a single line of a printer or display device.

XOFF, XON

For devices with limited data reception capabilities (inexpensive printers, for example), this message printing provision supports the “XON1 XOFF” protocol (except in model 2200 controllers). Devices using this protocol will automatically send an ASCII code “19” (‘XOFF”) if they need time to process data before receiving any more characters. The controller will cease message transmission upon receiving this code, until the receiving device sends an ASCII code “17” (“XON”). Due to timing differences within devices using the XON/XOFF protocol, it is recommended that this provision be tested carefully before use. A better practice would be to avoid its usage by timing the message transmittals from the controller to avoid overrunning the receiving device.

Special Purpose Registers for ASCII Message Transmitting

Register Number	Read/Write Access	Function
Register 12000	Write	Write Access Controls selection of active serial channel for message transmission. The Default is: store 0 = Channel A store (non-zero number) = Channel B
Register 12000	Read	Tests to determine if message transmission is complete. When `If REG_12000 = 0 Goto Next` is true, the out-going message is complete.
Register 12001	Write	Write Access Initiates transmission of message. Storing a number to Register 12001 causes the controller to transmit the data contained in that row of the Data Table to be transmitted. Storing a number higher than the number of rows present causes a software fault. For example, `Store 8 To REG_12001` causes the controller to transmit the ASCII data contained in row 8 of the Data Table via the currently active channel. (See Register 12000 Write access above).
Register 12001 to Register 12064	Read	Allows you to read the response, if any, from the device connected to the active channel. Registers 12001 through 12064 contain the characters 1 through 64 (as applicable) of any response received after a message transmission. The controller interprets the response as ASCII and stores the embedded numeric data as a succession of ASCII characters representing the digits of the number. The controller reads an ASCII carriage return (13) as the end of the response.

Messages stored in the controller's Data Table for transmission must conform to the following:

1. **Alphanumeric Data:** Text and numeric information is represented according to the standard ASCII code for data representation (see chart), using the numeric values 32 through 127. For example, the letter A is stored as the number 65 and the number 1 is stored as the number 49. Multidigit numbers are stored as a string of ASCII codes, where each digit is represented by a distinct ASCII code.
2. **Control Characters:** These special ASCII characters perform special functions within the receiving device and are represented by the numbers 1 through 31 and 128 through 255. For example, the ASCII code 13 is a carriage return command which, when sent to a printer, will cause the print head to return to the first column of the page. ASCII messages sent to a printer or display device are often terminated with the ASCII codes 13 and 10, which cause a carriage return, followed by a line feed (moving the print head, or cursor, to the next line. Other ASCII codes may have specific meaning for the device with which you are communicating.
3. **Controller REG-XXX References:** Often, it is desirable to include in a message the current value stored in one of the controller's registers. By including a number from 1001 to 1500 within your message, the controller automatically transmits an ASCII representation of the current value of register 1 (i.e.; counter 1) through register 500, in the message.

   NOTE: You cannot access register 128 using this method.

   It is possible to create messages such as: "The pressure in Vessel A is now" <register 10 value>, where the value in register 10 is printed immediately following the other text. Of course you would have first stored the pressure value into register 10, before calling for the message transmission. You can also print the values in other registers (register 501 and up), if they are stored in registers 1 register 500. For example, a log time may be derived from register 13002 (the millisecond-counter register) and, after your Quickstep program stores the value to register 100, you can print it by including the number 1100 in your message.
4. **Message Termination:** If a given message doesn't happen to exactly fit in the number of Data Table columns you are using, store the number 0 after the last valid character to be transmitted. The controller will not transmit the 0 and will act as a signal to the controller that the out-going message is now

ASCII Codes

Non-printing Codes		Printable Characters					
0	Nul (ends message)	32	Space	64	@	96	‘
1	SOH	33	!	65	A	97	a
2	STX	34	“	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL (rings device bell)	39	`	71	G	103	g
8	BS (backspace)	40	(	72	H	104	h
9	HT	41	)	73	I	105	i
10	LF (line feed)	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF (form feed)	44	,	76	L	108	l
13	CR (carriage return)	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI 4	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1 (XON)	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3 (XOFF)	51	3	83	S	115	s
20	DC4	52	4	84	T	116	T
21	NAK	53	5	85	U	117	U
22	SYN	54	6	86	V	118	V
23	ETB	55	7	87	W	119	W
24	CAN	56	8	88	X	120	W
25	EM	57	9	89	Y	121	Y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[	123	{
28	FS 6	60	<	92	\	124	|
29	GS	61	=	93	]	125	}
30	RS 6	62	>	94	^	126	~
31	US 6	63	?	95	_	127	delete