digelec

PROM & LOGIC
PROGRAMMERS

Received
5/91
Prom now 4.15

Dear User:

Thank you for buying DIGELEC.

As a result of software upgrades, Chapter 7 in your User Manual has become outdated. Please replace it with the enclosed revised Chapter 7.

Alex Abelson

Alex Abelson
Quality Assurance

CHAPTER 7

REMOTE MODE

7.1 INTRODUCTION

REMOTE mode is designed to enable you to control EP-804/5 from a remote computer, connected by means of the RS-232C port.

Communication between EP-804/5 and the computer is interactive, initiated when you send commands from your remote computer instructing EP-804/5 to perform various functions. EP-804/5, in turn, transmits back characters in response to these commands, advising you of the outcome of a function, or of general operating status.

This chapter serves as a guideline to enable you to incorporate EP-804/5 remote commands into your computer operating software.

7.2 SETUP

You may use REMOTE mode with any computer that has an RS-232C port. To activate it, proceed as follows at your programmer station:

a. Press <REM> and <ENTR>. The following message will be displayed:

   STANDARD REMOTE

b. To accept this, press <ENTR>.

c. Other types available include Intel MDS, Philips PMDS I and II. To select one of these, press <NEXT> to display it, and <ENTR> to select it. The interface parameters will automatically be adjusted.

Set the following transmission parameters at your computer station:

Type	Display	Baud Rate	Parity
Standard	STANDARD REMOTE	*	*
REMTYP	REMTYP CEL SLOW	9600	*
REMTYP	REMTYP CEL FAST	19200	*
Intel MDS	INTEL MDS REMOTE	9600	Odd
Philips PMDS I	PMDS 1 REMOTE	9600	Even
Philips PMDS II	PMDS 2 REMOTE	4800	Even

* Set to match EP-804/5

7.3 COMMUNICATION PROTOCOL

7.3.1 AT COMPUTER STATION

Most REMOTE mode commands consist of a single ASCII character, followed by a <CR>. The following keys at your remote computer also affect Model 804/5 operation:

<ESC> Aborts a command
<BREAK> Aborts data transfer
<CR> Signals completion of an action, both at programmer and computer stations

In cases where a command requires a parameter, it must conform with the following syntax:

XX...XXB, <CR>

where:
X = Any number of HEX characters, the last four of which are parameters (if fewer than four Xs, leading zeros are automatically added; if omitted, parameter is set to zero)
B = Command

7.3.2 AT MODEL 804/5

Model 804/5 responds to commands from the remote computer by transmitting back one of the following characters:

> Indicates successful activation of REMOTE mode, when followed by <CR>, or successful completion of function

F Precedes fault message. An error status word is sent as eight HEX characters (four bytes), where each bit represents an error code.

? Indicates unknown or invalid command

7.4 SELECTING A DEVICE TYPE

The first step in beginning operation is selecting a device type. This must be done before you insert a device in the socket.

To select the device type in KEY mode at your programmer, use initialization function 0. You may also select the device type in REMOTE mode at your computer, by sending command @.

If you are in REMOTE mode and attempt to execute a function before selecting a device type and inserting the correct device, the following message will be displayed at your programmer:

INSERT EPROM

Proceed as instructed, and press <ENTR> to acknowledge and continue.

7.5 THE COMMANDS

The commands described below are grouped as discussed in Chapters 4 and 5, and are listed alphabetically within each group.

7.5.1 SETUP FOR TRANSMISSION

Command	Function	Description
cccc@	SELECT EPROM TYPE	**WARNING!** Before using this command, make sure there is no device in the device socket! Sets device type, where cccc = EPROM device code, which you may enter without leading zeros;
xxA	SET DATA TRANSFER FORMAT	Sets one of the following data transfer format codes (xx): 00 Manual — 11 Motorola (S1) 01 HEX Space — 31 Ext. Motorola (S2) 03 BNPF — 12 Tektronix 10 HEX Intel — 13 MOS Technology 30 Ext. HEX Intel — 80 ceLink Binary

Command	Function	Description
D	SET ODD PARITY	**NOTE:** When altering the parity setting at your programmer, make sure that the setting at your computer coincides. Otherwise, communication may fail. Selects odd parity at programmer
E	SET EVEN PARITY	Selects even parity at programmer
G	SOFTWARE VERSION	Programmer transmits three-digit label indicating current software version
K	SET TWO STOP BITS	Sets two stop bits. Programmer responds by transmitting back two stop bits.
N	NO PARITY BIT	Selects no parity at programmer
aaaaW	SET ADDRESS OFFSET	Sets address aaaa, to be subtracted from all addresses transferred during data input through the serial interface, and to be added to addresses sent during data output
#	RESET	Resets hardware for serial interface and overload circuit

7.5.2 DATA TRANSFER

Command	Function	Description
I	INPUT PORT DATA TO RAM	Loads data into RAM from RS-232C port
L	INPUT EPROM DATA TO RAM	Loads data from programmed EPROM into RAM
O (Capital O)	OUTPUT RAM DATA TO PORT	Sends RAM data to computer
o (Lower case)	OUTPUT EPROM DATA TO PORT	Sends EPROM data to computer

7.5.3 RAM FUNCTIONS

Command	Function	Description
iiii<	SET RAM BEGIN ADDRESS	Sets RAM begin address to value iiii; default is zero
ssss;	DEFINE BLOCK SIZE	Defines block length ssss
_	CLEAR RAM	Fills RAM with a constant value, corresponding to erased state of device type, defined by address range character < (set RAM address) and ; (set block size)
C	RAM DATA COMPARED TO PORT DATA	Determines if discrepancy between data from port and RAM data (if yes, transmits F back to computer)
J	SPLIT	Performs split operation on device data on address range defined by characters < (set RAM address) and ; (set block size)
j	SHUFFLE	Performs shuffle operation on device data on address range defined by characters < (set RAM address) and ; (set block size)
S	RAM CHECKSUM	Calculates RAM checksum and transmits 16-bit result back to computer in HEX form xxxx. Useful for checking correct loading of data from computer.

7.5.4 EPROM FUNCTIONS

Command	Function	Description
bbbb:	SET EPROM BEGIN ADDRESS	Sets EPROM begin address to value bbbb; default is zero.
[	OUTPUT OF EPROM DEVICE TYPE	Requests EPROM device type, which programmer transmits back in form nnnn
]	EPROM AUTO MATCH	Instructs programmer to automatically identify device type of inserted EPROM
B	BLANK CHECK	Checks EPROM to determine if in erased state

Command	Function	Description
P (Capital P)	PROGRAM EPROM	Performs three programming stages: illegal bit test, programming, verifying; programmer transmits back F (fail) character if any stage fails, and > if entire process successful
p (Lower case)	PROGRAM 8751 SECURITY BIT	Programs security bit of 8751 family of single-chip microprocessors
R	REQUEST EPROM STATUS WORD	Programmer transmits back EPROM status word in form aaa/b/c or aaaa/b/c, where: aaa/a = Maximum EPROM address in HEX; (f < 1000H, three places are used; otherwise, four) b = Number of EPROM data bits (8) c = Erased state of EPROMs (0)
s (Lower case)	EPROM CHECKSUM	Calculates EPROM checksum and transmits 16-bit result back to computer in HEX form xxxx.
T	ILLEGAL BIT CHECK	Checks if programmed device can be programmed with RAM data
V	VERIFY DATA	Compares EPROM and RAM data

7.5.5 ERRORS

Command	Function	Description
&	SYSTEM STATUS MESSAGE	Programmer transmits system status message in form uWWWW/sXXXX/rYYYY/xZZ where: u = Unit (2000 = 804, 2100 = 805) s = Software revision number r = RAM size x = System error code
F	REQUEST ERROR STATUS WORD	Programmer transmits error status word in form xxxxxxxxx, where x = HEX character, and resets all bits to 0
X	REQUEST ERROR CODES	Programmer transmits error code in form XX, where X = decimal character strung together without separation
Y	REQUEST NUMBER OF PARITY ERRORS	Programmer increments internal counter if parity error found, and transmits this number to computer

7.6 ERROR HANDLING

7.6.1 ERROR STATUS WORD

Errors occurring during operation in REMOTE mode are recorded in the error status word, which can be retrieved by using the command F. The bits have the following designations:

Bit	Designation
31	Any error
30	Always 0
29	Always 0
28	Always 0
27	Always 0
26	Overrun error (character cannot be processed)
25	Framing error (no stop bit)
24	Always 0
23	Any EPROM error
22	Overload
21	No EEPROM type selected (command [)
20	EPROM device type number invalid (command @)
19	EPROM not erased
18	EPROM not programmable (only with function T)
17	EPROM data does not correspond with RAM data
16	EPROM not programmable (only with function P)
15	Any serial port error
14	Always 0
13	Always 0
12	Port data does not correspond with RAM data
11	Port checksum error
10	Port data word error
9	Port address error
8	Port character error
7	Any RAM error
6	Always 0
5	Always 0
4	Address range too small (only 3 V EPROMs)
3	RAM/EPROM address error
2	Always 0
1	RAM write error
0	Always 0

7.6.2 ERROR CODES

When it receives the command X, MOdel 804-5 generates error codes based on the bits of the error status word. The codes have the following meanings:

Code	Bit	Meaning
20	19	EPROM not erased
21	18	EPROM not programmable (only with function T)
23	17	EPROM data does not correspond with RAM data
23	16	EPROM not programmable (only with function P)
27	3	RAM, EPROM address error
28	2	Odd address for commands J and j
29	12	Port data does not correspond with RAM data
31	22	Overload
34	21	No EPROM type selected (only with function x)
34	20	Invalaid EPROM type number (only lwith function @)
35	29	Command j sent when no EPROM inserted
36	4	Address range too small (only 3 V EPROMs)
38	27	Command A sent without format code
40	30	<CLR> pressed
48	26	Overrun error (character cannot be processed)
48	25	Framing error (no stop bit)
63	1	RAM write error
81		Parity error
82	11	Port checksum error
84	8	Port character error
91	9	Port address error
99	28	Timeout error during data transmission in ceLink format

Following are examples for EPROM type 2732. The highest address is 0FFF.

a. No address inputs (default values)
   RAM begin address: 0000 RAM end address: 0FFF
   EPROM begin address: 0000 EPROM end address: 0FFF

b. Additional input of block size 0A00 (command 0A00;)
   RAM begin address: 0000 RAM end address: 09FF
   EPROM begin address: 0000 EPROM end address: 09FF

c. Additional input of RAM begin address 1000 (command 1000<)
   RAM begin address: 1000 RAM end address: 19FF
   EPROM begin address: 0000 EPROM end address: 09FF

d. Additional input of EPROM begin address 0600 (command 0600:)
   RAM begin address: 1000 RAM end address: 19FF
   EPROM begin address: 0600 EPROM end address: 0FFF

e. The following additional inputs cause address errors:
   0A01; EPROM end address: 1000 (exceeds EPROM maximum address)
   0601: EPROM end address: 1000 (exceeds EPROM maximum address)
   1601< RAM end address: 2000 (not available for 8K RAM)