Auxiliary Software for Memory-Manager 3.0

MM-EPROG

Function Description

Version 1.59

system-99 user-group

Table of Contents

Application

MM-EPROG is the first of the Auxiliary Software programs specially written by Ch. Winter for Memory Manager. It is intelligent EPROM programming software witten for the EPROM programming device the EPROG 27011 that, at a price of approximately DM 200,- is available at electronics mail-order outlets.

This programming device allows one to program, read, and verify EPROMs from 2Kbytes to those on the order of a MegaByte. The EPROG 27011 uses two modes.

On the one hand, the so-called standalone mode makes it possible to produce a 1:1 direct copy of an EPROM without a computer. Also available is an empty test or the comparison of the contents of two EPROMs. With two rotary switches, the EPROM type and mode can be started and stopped. Two LED displays show, through a key press, the operation mode of the EPROG 27011.

The second mode of operation is the 'Computer mode'with which we will mainly discuss from now on. Here it is possible to pick out certain parts of the contents of the EPROM to read which can be edited and saved to program another EPROM.

Consoles with Altered Crystals

After the program is selected the program software first does a test of the clock frequency. This is necessary because baudrate of the card interfaces are based on the calculations of a 12 MHz CPU clock and the interface chips use this clock frequency. An altered clock leads to incorrect baudrates. The worst case means that the commands to the EPROG 27011 are not correct or are falsely interpreted. In this case they are known in most circles as 'act of God' EPROMs.

In the case where, while choosing the programming software, an incorrect clock frequency is detected this will be displayed and the program will wait until the correct clock frequency value is entered or a key is pressed. In the case of a keypress, you are returned back to the main program. As soon as the correct clock frequency is detected you will be sent to the main menu of the programming software. Many of the routines of Memory Manager are used and many points are the same and identical to the main program. So, the functions are executed by pressing the individual character keys.

The main menu of the program sections gives a glimpse of the information necessary to program an EPROM. The data in this screen is transferred to the programming buffer after selection and used for work.

Adjusting the EPROM Parameters

By repeatedly pressing the 'T' key you can select the EPROM type through all of those that can be worked on by the EPROG 27011. The different types appear twice, the reason being that during the course of development and design changes there are different electrical current values for them even though the storage capacity has remained the same. These 'modern' types are designated with an 'A' or 'a'.

Programming Voltages

The EPROG 27011 puts three programming voltages at your disposal 12.5 Volts, 21.0 Volts and 25.0 Volts. The programming voltage can be adjusted by pressing the 'U' key. This adjustment needs special attention since it can lead to unfortunate changes in the EPROM. For example the 2732 types must be programmed with 25.0 Volts and the 27011 types with a maximum of 12.5 Volts. Too high of a programming voltage results - in all cases - in the destruction of the EPROM. Too low programming voltage results in the EPROM not being programmed. It is wise that with an unknown type of EPROM that one should start with the lowest voltage and upon finding

programming errors then go to the next higher voltage value. An alternative is to use the programming voltage 'AUTO'. Here the EPROG 27011 will choose the programming voltage based on the EPROM type. It should be noted that the CMOS type are not contain in the list and the incorrect programming voltage might be chosen.

Programmer Algorithms

In order to adjust the programming algorithm one can press the 'A' key. The EPROG 27011 has four programming algorithms available. This allows the EPROM to be programmed in the manner that the manufacturer suggests.

Standard programming

With standard programming each byte is 'burned' with 50ms of the programming voltage. This is repeated with every byte until all of the bytes are programmed. The programming time for a 4 Kbyte EPROM is 5 minutes.

Intelligent Programming

The EPROMMER also is aware of the intelligent algorithms of INTEL, AMD and Fujitsu. This turns on something else that is known as the so-called 'Quick pulse mode'. Then each byte is burned with the programming voltage for only a 1 ms, and immediately read to see if the information in the memory cell is correct. If that is not the case, the process is repeated up to 50 times until the information is put into the EPROM. For security, the programming voltage is turned off in the so-called 'after pulse' for a few milliseconds.

The different programming algorithms differ by the number and duration of the programming impulses and the afterpulses. Since programming in the standard mode lasts a relatively long time, particularly with highly integrated types, they can be destroyed by fast programming data overruns. As a small decision making aid, the following adds some clarity to the described type list.

Danger!

In order for the programming of an EPROM to be as flexible as possible the program does not pay attention to whether the selected EPROM type, programming voltage, and algorithm go together.

So it is possible to program an INTEL 2732 EPROM using 12.5 volts. It may very well work. It is also possible to burn a 27512 with 25.0 volts. This is sure to cause the death of the EPROM.

Banked EPROMs

Depending on the selected EPROM type i.e., the two cases of the 27011 and 27513 allow you to choose the bank of the EPROM. These two EPROMs are not linear addressable but divided into banks which makes the same address range available more than once. By pressing the 'B' key and after the input, the desired bank can be chosen. Depending on the type of EPROM selected the inputs of 0-3 (27513) or 0-7 (27011) are possible.. In all other types the bank input is ignored.

Buffer Start Address

With 'S' the start address in VDP buffer, from which the data is programmed, and read to compare and/or written into is selected. The end function with 'E' allows the setting of the end of the data buffer. The difference in the values (plus 1) gives the count of the number of bytes to the program end or read end. These can be altered with 'E' and 'S'.

Offset Address

The offset in the EPROM itself, the relative EPROM start address, can be chosen with 'O' for the SLAVE socket or with 'CTRL-O' for the MASTER socket. Beginning with the offset address, as many bytes are programmed or read as was specified in the byte count. In this application, as an example the 27128 where one wishes to from the bytes from >2000 to >20FF, the offset is must be >2000 and the byte count >0100. The byte count only depends on the input of the Start- and End values and can not otherwise be adjusted. When the start is at >0000 and the end is at >00FF that means that >0100 bytes can be read, compared, or programmed.

Mode: 8 or 16 Bit

Since our dear TI99/4a is a 16-Bit computer, it is possible to divide the buffer contents into two EPROMs each with even and odd addresses.

The switching is done with repeated pressing of the 'M' key.

With the display of 8-bit programming of the EPROM, Lowbyte programs only the even bytes and Highbyte only the data at the odd addresses.

The number of bytes that can be programmed is adapted in either case.

Here is a general reference of the different key presses between the Master- and Slave-EPROM:

All commands concerning the Slave-EPROM are placed with a capital letter. If one wish to reach the EPROM that is in the Master socket one must press CTRL key at the same time.

This does not function during programming since the EPROMMER can only program the SLAVE socket.

Reading EPROMs

After setting the Start- and End address, in order to read the contents of an EPROM and place them into the VDP buffer one must press the 'R' key for read for the Slave socket or 'CTRL-R' should one wish to read the Master socket. The software acknowledges this with 'Lesen' (Read) in that status line. The time required depends on the values. It can amount up to a minute. When the read operation is finished, 'Daten korrekt gelesen' (Data correctly read) will be displayed in the status line. This message is displayed until >ENTER< or another valid command key is pressed. From the EPROM start address, as many bytes as were read and 'Anzahl Bytes' (Number of Bytes) are placed in the VDP buffer beginning with the specified start address up to the end address.

Nothing now stands in the way of going back to work with the main program of the Memory Manager.

In the rare event of a message 'Fehler beim Lesen' (Error during Reading) appearing it is possible that EPROG 27011 had an error in reading generated by the clock. In this rare case, it can be repaired by pressing the 'RESET Key' on the EPROMMER.

Programming

For programming an EPROM, proceed exactly as you did while reading an EPROM: set the Start- and End address accurately and pay particular attention to the programming voltage and algorithm!!!

As we have discussed many times before, by setting incorrect options it is possible to send the EPROM directly to Heaven (or Hell?).

When everything is properly set, press 'P' once to program and when you are asked if you are sure you want to continue, answer by pressing 'J'. All other keys lead back to the main menu. There is

no purpose in pressing CTRL-P, since you are not allowed to program the Master socket. During programming, the message 'Programmieren' (Programming) appears in the status line. At the end of the programming operation, hopefully the message 'Daten korrekt programmiert' (Data correctly programmed) will appear.

Should the message 'Fehler beim Programmieren!' (Error during Programming) appear, there can be several different causes. The simplest case is if the EPROM is not empty and still contains data. In that case, you will need a so-called EPROM erasing device (obtainable seperately).

If the EPROM was already empty, it is possible that a higher programming voltage is needed. Considering that there might be destruction, raise the programming voltage slightly. It is also possible that changing the programming algorithm will lead to the desired success. Before one gets the the very end, and goes to the standard programming mode, one should try all of the fast programming modes for this EPROM. If you still do not reach the desired results, the EPROM is likely defective. This is sometimes recognizable if the character patterns can be read after EPROM was erased and is read. In this case, help for the patient comes too late (rest in peace...)

Empty Test

The third function of programming is the 'Leertest' (Empty Test) it is selected with the 'L' key or likewise 'CTRL-L' and it works independently of the Start- and End added or the offsets. The empty tests tests the entire EPROM for >FF contents (>FF = empty). The VDP buffer is not changed. The two messages 'EPROM leer' (EPROM empty) and 'EPROM nicht leer' (EPROM not empty) need no further clarification.

Data Comparison

In case you wish to compare the contents of the EPROM with those of the VDP buffer for security this can be done with the key function 'V' (or 'CTRL-V' for the Master). Everything between the borders of the buffer are checked completely (!!!) that the EPROMMER sent and compared with the data that is in the EPROM. The messages 'Daten identisch' (Data Identical) and 'Daten nicht identisch' (Data not Identical) refer to the entire tested range. The exact identification of the faulty bytes is not yet possible.

Check Sum Test

One of the possible wishes starting in Version 1.33 was checksum testing of the entire EPROM contents. This occurs by pressing 'C' for the Slave-EPROM or 'CTRL-C' for the Master-EPROM. After pressing the key, the message '* Prüfsummenermittlung' (Checksum Test) is displayed in the status line. Again, depending on the Start- and End address, this can take a long time during which there is nothing on the screen. Note: the checksum is formed from all of the data within the set range. In order to do checksum of 27512 EPROMs one must set a starting value of >0000 and an ending value of >FFFF. The checksum is computed byte for byte without an overflow. In order to do checksum of the buffer contents the FCTN-C be pressed once. The buffer contents are not altered.

Help Screen

All of the functions discussed so far can be found with '?' in a small help window that can be used as a 'Quick reference' memory aid.

Auxiliary Functions

An auxiliary screen is possible in which to set various parameters. As soon as the help screen has been chosen with the 'Z' key you can set the parameters for various special applications.

Interface Port

One of the most important settings is the port which interfaces with the EPROG 27011 and to which it is connected. It is possible to use any of the four available RS232 ports. This can be indicated with the 'P' key. You must press it as many times as it takes to get to the display of the desired port.

All of the other transmission parameters of the EPROMMER are set.

The baudrate is set at 9600 Baud, data bits are set to 8 and the parity bits are off.

These values have turned out to be the most favorable for this application. Should you wish to have other parameters, please contact me.

Programming Mode

During the setting of the program mode with the 'M' key you are presented with the following possible choices: ALL, Lowbyte or Highbyte. What does this mean? The TI 99/4a is, as you hopefully know, is a 16-Bit computer. It naturally follows that is has a 16-bit wide data bus. Which it does. Since, however, there are no EPROMs with a this wide bus, two EPROMS are used one for bits 0-7 and another for data bits 8-15. It was penciled-in that the so-called console ROM was built into the range of >0000 - >1FFF. If you want to invent a new operating system for the TI with this EPROM package, then you must switch program modes. In the 'Lowbyte' mode all of the odd bytes are burned into the EPROM. Then, all of the even bytes can be read and burned. In normal operation, however, 'Alle' should be indicated.

Alternatively the display of the command strings on the screen during programming is actually a leftover from the development period of this program. With the use of 'S' which can be switched off it is possible to indicate which will be sent as a command sequence to the EPROG 27011. Thus any errors in the settings are easily discovered.

Actually, there is already a small decision making aid above once the type list has been seen. The list was composed from data sheets and information from various manufacturers and adapted to the format of the EPROG 27011 manufacturer. The first column indicates the type of EPROM as printed on the EPROM. The second value is the type which can be adjusted with 'T'. The third indicates whether the EPROM can be programmed with standard Mode. If applicable it is indicated by a '*'. If the EPROM can be programming in intelligent algorithm mode, this is indicated in column four. The last row gives the highest programming voltage. Stepping through the tables other manufacturers is done by pressing BACK and pressing it again returns you to the main menu.

Configuration File

Once you have set the program to your own special application it is possible for you to find the program 'work ready' when you use it next. After setting the parameters of this part of the program, choose the Configuration menu from the main menu. This is described under the instructions in the main menu under 'Configuration'. It is naturally quite practical to save the values and settings for programming several EPROMs.

Through the concept of auxiliary software it is actually possible to read the configuration file of other auxiliary software. In this case, the program recognizes the false data as an 'ungültiges Parameterfile' (Invalid Parameter File) and ignores it. The default values now appear on the screen. You can now produce your own configuration file using this pattern as an example.

Error Messages

Except for the messages like 'EPROM nicht leer' (EPROM not Empty) or 'Fehler beim lesen' (Error while Reading) there is one other message that does not fit into a pattern.

As soon as the 'Timeoutfehler beim Lesen' (Timeout error while Reading) appears in the status line it is an indication that the TI 99/4a and the EPROG 27011 do not understand each other and an error has arisen during data exchange. The simplest reason could be, for example, that you simply forgot to attach the EPROMMER to the TI 99/4a

Another possibility is that the DIP switches on the EPROG 27011 are not set correctly or the wrong port has been chosen. Thus, a reading operation from the TI 99/4a to the EPROMMER is broken off which follows the action after about 2 seconds. Then, the hardware error is given.

A further error message that is not specifically related the action of the EPROMMER connection is the result of setting incorrect buffer Start- and End values. Thus, the values prevent the data that is written in the buffers to be written into the system range and the program can crash. The error message 'Puffergrenze' (Buffer Border) is confirmed with any key but not implemented. The monitoring of the buffer borders is done during reading, comparing, and programming but not during the empty test or checksum calculations since these work independently (with checksum) in the same range as the EPROM, for example as in a 27512 from >0000 to >FFFF.

Connection from the EPROG 27011 and the TI 99/4a

The connection does not have to be made via a shielded cable but should be less than about 5 meters to one of the four serial ports. It is most practical to use RS232/2, the second port which is most likely not used by the user. That way, you can use a permanent connection for a modem or printer connected to RS232/1.

RS232/1	RS232/2	EPROG 27011	Signal
Pin 8	Pin 12	Pin 2 CTS	Handshake In
Pin 7	Pin 7	Pin 5 GND	Signal ground
Pin 20	Pin 19	Pin 4 RTS	Handshake Out
Pin 3	Pin 16	Pin 1 RxD	Data in
Pin 2	Pin 14	Pin 3 TxD	Date out

DIP-Switches

For normal operation the DIP-Switches must be set on the EPROMMER circuit board. For this software the following are valid:

	1	2	3	4	5	6	7	8
On				*	*			
Off	*	*	*			*	*	

Setting the DIP-Switches should take place before the EPROMMER is turned on, otherwise damage could result. The setting of DIP-Switch 8 is insignificant in the stand-alone mode since the computer is not needed.

The operation of the EPROG 27011 was only tested on an original TI-System with an original interface card. Also only the EPROMMER with the version number 'UPVAV10 4/87' was available. It is possible that other EPROMMERs or other interface cards can be used but the function of these cannot be guaranteed.

Finaly, let me state again that neither Texas Instruments nor I as the author of this program are liable for any damages that occur from the use of this program and these instructions.

Quick Reference for MM-EPROG

A ....Choose program algorithm.

B....................Set the Bank of the EPROM (only 27011 & 27513)

C....................Checksum test (Data in EPROM)

c....................Checksum test (Data in buffer)

E....................Set End Address in Editor buffer

L....................Empty Test

M....................Program Mode (8/16 Bit)

O ....EPROM-Offset (First given Address in EPROM)

P....................Program the EPROM

R....................Read the EPROM

S ....Set the Start address in the Editor buffer

T....................Choose the EPROM type

U ....Choose the programming voltage

V ....Compare the data in the EPROM and the buffer

Z....................Auxiliary functions

P ....Set theRS232 ports

....................Version number and date

?....................EPROM-Information

FCTN-7..........Display of the key presses with HELP

FCTN-9..........Leave the program section with BACK

FCTN-=..........Initialize the EPROMMER and RS232

The commands C, L, P, R und V are only for the Slave socket on the right.

The Master socket on the left is accessed with FCTN key (except for programming) Programmieren).

Reaching other program parts of Memory Manager with CTRL and the designated key from the main menu:

CTRL-E ....Editor

CTRL-T ....Transfer

CTRL-K.........Configuration

CTRL-D.........Print

CTRL-I ....I/O Disk