

nemon

Communicating with
an embedded system

Reference Guide

nilsen elektronikk as

Jan Erik Nilsen

May. 18. 2000

nilsen elektronikk as
Gml. Drammensvei 12B
N-1369 Stabekk
Norway

Tel: +47 67 58 31 62

Fax: +47 67 58 97 61

<http://www.nilsenelektronikk.no>

TABLE OF CONTENTS

LICENSE AGREEMENT / DISCLAIMER.....	2
OVERVIEW	3
THE COMMUNICATION PROTOCOL.....	4
Command format.....	4
Answer format.....	6
THE MAGIC STRING.....	7
STARTING nemon	8
COMMANDS.....	9
THE RS485 OPTION.	10
SCRIPT FILE FORMAT	11
APPENDIX.....	12

LICENSE AGREEMENT / DISCLAIMER

Copyright © nilsen elektronikk as

The **nemon** with documentation are properties of **nilsen elektronikk as**, Norway. **nemon** is free software; you can use it, redistribute it and/or modify it under the terms below. By using, changing or redistributing the software, you accept the conditions below:

1. You are not allowed to remove or modify this copyright notice and License paragraphs, even if parts of the software is used.
2. The improvements and/or extentions you make **shall** be available for the community under **this** license, source code included. Improvements or extentions, including ports to new operating systems, **shall** be reported and transmitted to Nilsen Elektronikk AS.
3. You must cause the modified files to carry prominent notices stating that you changed the files, what you did and the date of changes.
4. You may **not** distribute this software under another license without explicit permission from Nilsen Elektronikk AS, Norway.
5. This software is free, and distributed in the hope that it will be useful, but **WITHOUT ANY WARRANTY**; without even the implied warranty of **MERCHANTABILITY** or **FITNESS FOR A PARTICULAR PURPOSE**. You **SHALL NOT** use this software unless you accept to carry all risk and cost of defects or limitations.

The software is provided «as is» and without any express or implied warranties, including, without limitation, the implied warranties of merchantability and fitness for a particular purpose. The software should not be used within Life Support Systems. Life Support Systems are equipment intended to support or sustain life and whose failure to perform properly used in accordance with instructions provided can be reasonably expected to result in significant personal injury or death.

OVERVIEW

nemon consists of a communication protocol and a communication program running on a PC. A PC comport is connected to the embedded system. The main purpose is to provide an easy way to load a program into writeable memory of the target. A fast binary protocol is used. With **nemon**, an arbitrary memory type of the target can be accessed. **nemon** can:

- Erase sectors of a FLASH memory.
- Load a program from a Motorola S or Intel Std. file into the target.
- Display, once or live, memory contents, HEX / ASCII, from the target.
- Modify writeable memory of the target.
- Break / Enter the target application program.

Reading, writing and control of the target system is carried out by a boot-monitor contained in write-protected memory. The monitor is small, typical 250-1K bytes. By including another (optional) monitor in the application program, **nemon** can also be used during application program debugging.

nemon consists of the program **nemon**, with ANSI C source code and documentation. The communication protocol is described, and some examples included.

The **nemon** protocol provides commands for control, read and write.

- Control commands are: Reset the target, Jump to a given address, Enter an application program.
- Read commands can read 1-32 bytes from the target at a memory area starting at an arbitrary address in an arbitrary memory map.
- Write commands can write 1-32 bytes to the target at a memory area starting at an arbitrary address in an arbitrary memory map.
- Another command can erase one given sector of a FLASH memory.
- Another command is user defined, and may take user defined actions.

nemon provides the following user commands:

- Display the contents of memory as ASCII, hexadecimal bytes or words. It can be done once, or repeating to get a live image. The user is asked for a range and memory type. Memory type may be RAM, serial EEPROM, FLASH or anything. The monitor is responsible of accessing the respective type.
- In a similar way, contents of writeable memory can be changed.
- By transmitting a sequence of 10 «magic» bytes to the application program, it can be ordered to enter the boot monitor. The detection of the «magic» bytes are done by means of a simple state machine in the UART handler of the application program.
- Sectors of FLASH memory in the embedded system can be erased by the boot monitor.
- Program can be read from a Motorola S or Intel file and transmitted to the boot monitor.
- By giving a command, the boot monitor can enter the application program.
- **nemon** includes a simple terminal emulator.
- For convenience, **nemon** can change quickly between two baud rates, one for the boot monitor and another for the application.
- To simplify program loading and other operations, commands may be read from a script file.

nemon supports both 16 and 32 bits addresses.

At system reset, the boot monitor gets control first. If it finds a signature at a given address in the application program space, control is immediately given to the application program.

The application program enters the boot monitor when 10 «magic» bytes are received. How likely is this to happen by accident?

10 bytes are 80 bits. 2^{80} different values can exist. In 10^9 years from now, the sun goes nova. If random bytes are received at a rate of 115200 baud, $3.6 \cdot 10^{20} = 2^{68}$ bytes can be transferred during this time, i.e. 2^{68} different values. The probability of a match is $1/4000$.

THE COMMUNICATION PROTOCOL

The protocol is binary and very efficient. The implementation is limited to one program module, and easy to change if the user wants to do so, e.g. if an ASCII protocol must be chosen.

Command format

The PC host sends binary records with the following format:

[0xFF, 0xFF<rs485addr>] <tag><crc><len><addr><data>0x5A

0xFF	2 bytes	RS485 preamble (RS485 option only)
<rs485addr>	1 byte	RS485 address. (RS485 option only)
<tag>	1 byte	Command / identifier
<crc>	1 byte	Byte Sum of the <addr><data> field.
<len>	1 byte	Length of the <data> field. Max: 32.
<addr>	w bytes	Address field, MS first. w is 2 or 4, given by <tag>.
<data>	n bytes	Data field. n is given by <len>
0x5A	1 byte	For re-synchronisation in case of transmission error.

The **<rs485addr>**

If, and only if, the command line option rs485 is chosen, this field is included. The RS485 multidrop address provide a way to select one target. The RS485 option is explained later.

The **<tag>**

<tag> & 0x0F is command:

- 0x00 Reset target. **<addr>** is ignored. **<len>** is 0.
- 0x01 Jump to the address given in the **<addr>** field. **<len>** is 0.
- 0x02 Write **<data>** of length **<len>** to memory starting at address **<addr>**.
- 0x03 Read data of given length starting at address **<addr>**. The length is given in **<data>**, 1 byte. The **<len>** value is 1.
- 0x04 Erase one sector. The sector index is **<addr>**. **<len>** is 0.
- 0x05 Enter the application program. **<addr>** is ignored. **<len>** is 0.
- 0x06 User defined command. **<addr>** is ignored. The user may enter **<data>** of length **<len>** to the receiver of user defined commands.

<tag> & 0x10 is address width, w, see above:

- 0x00 2 bytes
- 0x10 4 bytes

<tag> & 0x60 is memory type, also called map. It describes which algorithm to be used for a given memory. The user is free to define the use of different values, and the following is just a suggestion:

- 0x00 map 0: RAM
- 0x20 map 1: FLASH memory
- 0x40 map 2: Serial I²C memory.

<tag> & 0x80 is not used.

The unused tag bit and the unused commands can be used to project specific extentions. This way, the nemon protocol can be adapted for system communication.

Answer format

After giving a command, **nemon** waits until acknowledge is received. Therefore, the boot monitor does not need to be interrupt driven.

Four different acknowledges are defined:

' * '	0x2A	OK.
' # '	0x23	ALIVE. Returned after: <ul style="list-style-type: none"> • Unknown or invalid command or parameters are received by the target. • The boot monitor is entered from cold start or from the application program . • An application monitor is entered from the boot monitor.
' - '	0x2D	ERROR. CRC error or Invalid length
' ! '	0x21	Parameter invalid or out of range.

Answer from target, RS232:

The target responds binary data with the following format:

[<data>] <ack>

<data>	n bytes	Respond to Read Data command, see above.
<ack>	1 byte	Acknowledge. See above.

Answer from target, RS485:

0xFF, 0xFF, 0xFE <ack> <crc> <len> [<data>] 0x5A

0xFF	2 bytes	RS485 preamble
0xFE	1 byte	nemons RS485 address.
<ack>	1 byte	Acknowledge. See above.
<crc>	1 byte	Byte Sum over the <data> field.
<len>	1 byte	Length of the <data> field. Max: 256.
<data>	n bytes	Respond to Read Data command, see above.
0x5A	1 byte	For re-synchronisation in case of transmission error.

Reserved RS485 addresses:

0xFF Used for preamble.

0xFE **nemon**. This address is always used in answer from target.

0xFD Broadcast. Only target with address = 0 is allowed to answer.

THE MAGIC STRING

When the application program receives the magic string, it jumps to the boot monitor and remains there until a execute, jump or reset command is given.

The magic string was generated by using a highly random function:

```
static char *magic_string = "\x30\x91\x0C\x0C\x14\xF7\x05\xC7\xE9\x05";
```

The string can be tested by the receive interrupt handler. If interrupt is not used, the string can be tested for in the getCharacter function by calling the function below:

NB: The magic string is not a command. No acknowledge should be given.

```
/*===== nemonCheckMagicString =====
*
* Purpose:      To be called by the Receive Interrupt Handler
*               with the current received byte.
*               This function checks the input for magic string.
*               The function or macro «enterBootMonitor()»
*               must be supplied.
*
* Input:       current byte
*
*/
void nemonCheckMagicString(int ch)
{
#define MAGIC_STR_SIZE 10
static unsigned char *magic_string =
"\x30\x91\x0C\x0C\x14\xF7\x05\xC7\xE9\x05";
static unsigned char *msptr;

if (!msptr) msptr = magic_string;
if (*msptr == (unsigned char)ch) {
    msptr++;
    if ((msptr - magic_string) == MAGIC_STR_SIZE) {
        msptr = magic_string;
        enterBootMonitor();
    }
}
else {
    msptr = magic_string;
    if (*msptr == (unsigned char)ch) msptr++;
}
}
```

If the **rs485** option is used, the multi-drop address (1 byte) is transmitted first. The function above can be modified to include the RS485 address and thus be addressable.

STARTING nemon

nemon is started from the DOS prompt:

```
nemon [<opt>].. [<filename>]
```

Options are preceded by - or /

f	Fake. Pretend, don't use any comport. For demo.
cb<comparam>	Comport parameters for the Boot Monitor
ca<comparam>	Comport parameters for Application Monitor
e<b l>	Endian, Big or Little. Affects command 'D', 'L' and 'C'. Displays multi-byte values with different byte order. Big Endian is Most significant byte at lowest address. Default is Big Endian.
a<w l>	Architecture, w: 16-bits, l: 32-bits. Affects command 'D', 'L' and 'C'. Displays multi-byte values with appropriate number of bytes. Default is 'w'.
rs485	Multi-drop mode. Explained later.

The format of <comparam>:

```
[<n>:][<b>[,<d><p><s>]][/ <h>]
```

<n>	1..4	2	Comport number.
	50..115200	38400	Baudrate.
<d>	7..8	8	Data bits. Must be 8
<p>	N,O,E	N	Parity.
<s>	1..2	1	Stop bits.
<h>	N,H,X	N	Handshake. Dont use.

<filename> is the file name of a script file. Default file extension is «.in». Commands can be read from a script file. Commands and script file format are explained later.

Command line options can also be read from the script file. If the first character of a line is - or /, the line is supposed to contain one option. The command line is read after the file, i.e. command line parameters overwrites parameters from the file.

When **nemon** starts, a prompt appears:

```
--:
```

There are two prompt types.

```
--:      Comport parameters for boot monitor is used.
==:      Comport parameters for application monitor is used.
```

COMMANDS

Q,q	Quit.
R,r	Reset target. The target will jump to a cold_start_entry or starve the watchdog.
F,f	Load data from file into the current map of the application. The user is prompted for a file name. Default file extension is «.hex». Motorola S and Intellec file formats are detected automatically.
S,s	Send the «magic» string. The target will enter the boot monitor.
X,x	Jump to the entry point of the application program.
G,g	The user is prompted for an address, and the target makes a jump to the address.
d	The user is asked for «from» and «to» address, and the area in the current map is dumped on the screen. Default «to» = «from» 0xFF. Hexadecimal bytes and ASCII are used.
D	Same, but 16 or 32-bits values and ASCII are displayed. Word size and endian can be specified in the command line, see options.
l	The user is asked for «from» and «to» address, and the area in the current map is displayed live on the screen. Default «to» = «from» 0xFF. Maximum range is 256. Hexadecimal bytes are used.
L	Same, but 16 or 32-bits values are displayed. Word size and endian can be specified in the command line, see options..
c	The user is asked for an address, and the byte at the address in the current map is displayed. The user is asked to change the byte. + and - can be used to step forward and backward, respectively.
C	Same, but 16 or 32-bits values and ASCII are read and written. Word size and endian can be specified in the command line, see options..
T,t	NB: Not available if rs485 option is chosen. Enter terminal mode. A simple ASCII terminal for communication with the application program.
M,m	The current memory map number is printed, and the user is asked for a new map number. The map number affects command F,D,C,L,I,O and E.
A,a	Change comport parameters to the Application Monitor Comport parameters.
B,b	Change comport parameters to the Boot Monitor Comport parameters.
W,w	Set wait between commands to the application. The user is asked for a value given in milliseconds. If the application processor is busy and heavy loaded, high UART traffic may bust it. A wait time simply appends silence after issuing a command. NB: The F,D,L,I, and O commands usually transfers 16 bytes at a moment, and the wait is not made until all 16 bytes are transferred.
I,i	Initialise the application memory. The user is asked for «from» and «to» addresses and a value. The value is written to the area of the current map.
O,o	Output data from the application program to a file. The user is asked for «from» and «to» addresses, file format and file name.
E,e	Erase sectors of a FLASH memory. . The user is asked for «from» and «to» sector numbers. Sector size and range is user dependent. The «erase sector» command is repeated n times by nemon , (n = to-from+1).
U,u	Used defined command. The user is prompted for data to enclose the command. One string of hexadecimal bytes of an arbitrary length can be given.
Z,z	nemon will sleep a given number of seconds. The user is asked for the number. This command is useful in scripts when the target reset recovery time is long.
Y,y	BREAK communication. The line is held in «ON» state for 1 second. Some targets are equipped with HW detection of BREAK for HW reset.
N,n	NB: Only available if rs485 option is chosen. The current RS485 multi-drop address is printed, and the user is asked for a new value. Range: 0..255.

THE RS485 OPTION.

This option was included for using **nemon** in a multi-drop network. The following hardware requirement must be satisfied:

- A RS233/RS485 adaptor must be inserted between the PC comport and the RS485 network.
- The adaptor transmitter must be controlled by RTS or DTR. (Transmit Enable). The transmitter impedance is low when RTS or DTR is HIGH, vica verca.
- The adaptor receiver must always be enabled. **nemon** needs to read what it transmits in order to switch RTS and DTR off after transmitting the last byte.
- The RS485 line should be biased such that all connected receivers reads «idle line» when all transmitters are in high-impedance state. Otherwise, false data can appear when the impedance is high.
- The protocol ensures that only the addressed target will answer, and there will be no collisions, with one exception: At power-up reset. This will, however, not harm. It can be avoided by introducing a dummy wait, of length depending on the multi-drop address.

The purpose of the two preamble bytes is to recover from false data which can be made when the impedance is switched. Therefore:

- First, leading bytes $\neq 0xFF$ must be ignored.
- Second, bytes $== 0xFF$ must be ignored.
- Then «real» data follows.

SCRIPT FILE FORMAT

Commands may be input from a script file. The file format is:

```
#
# Comments
#
characters are read as is
#
# Trailing ' ' and '\t' are ignored.
# '\n' and '\r' are always ignored.
# '!' is changed to '\r' (The enter key)
#
Fprog.hex!      # comment
#
# File input:  prog.hex <ENTER>
#
```

When the user is asked for an address, file name etc., <enter> and <space> are legal terminators. If <esc> is used, the command is exited.

If the default comport parameters does not fit, another parameters must be given in the script file. Command line optional parameters must be preceded with - or / as the first character of a line.

The following script updates the program of an application:

```
-cb1:115200      # boot monitor: comport 1, 115200,8N1
-ca57600        # application monitor: comport 2, 57600,8N1
S               # enter the boot monitor
Z2!             # recover from reset. (slow RAM check)
M2!             # set memory map = 2
E0 3!          # erase sector 0..3
Fprog!         # load file «prog.hex» into the current map
X               # enter the application program
Q               # quit
```

APPENDIX

What happens when a 'X' (execute) command is given:

- The command record «Enter the application program» is given.
- If the target boot monitor cannot execute the command because no application program is loaded, '#' is returned. Otherwise, '*' is returned.
- If '*' is received, **nemon** changes to application comport parameters.
- If '*' is received, and the next received character is '#' (application program has started), the '#' is not displayed.

What happens when a 'S' (send magic string) command is given:

- **nemon** sends the magic string and changes to boot monitor comport parameters.
- Target receives the magic string and enters the boot monitor. It may transmit a «hello» string, which should be terminated with '#'.
- **nemon** waits for a '#' character for about 3 seconds. Characters preceding the '#' are ignored. As soon as the '#' is received or the time is out, the next step is:
- **nemon** transmits an invalid command (cmd = 0x0F).
- Target answers '#'.

AT RESET

At hardware reset, the boot monitor gets control. Control can be given immediately to an downloaded application program. The following suggests a method:

The boot monitor checks the content at address PADDR. If the value PMAGIC is found, the next memory word is expected to contain the entry address of the application program, which is entered immediately. Otherwise, the control remains in the monitor.