...the broadest narrowband money can buy

RACOM
RADIO DATA NETWORKS

User applications

Downloading firmware

version 7.57
2/15/2008

RACOM s.r.o. • Mirova 1283 • 592 31 Nove Mesto na Morave • Czech Republic
Tel.: +420 565 6595 11 • Fax: +420 565 6595 12 • E-mail: racom@racom.cz

www.racom.eu

Table of Contents

1. Download firmware into CU - introduction

Downloading is the process of inserting a firmware program module into a radio modem.

The CU is a common name for the Communication Unit of the MORSE system (Communication Unit), which can be the MR400, MR160, MR25, MG100, MC100, MR25ET, MCM302, MR900, sw WALRUS, sw MORCE. The **Firmware** (fw) is that part of the software, which is loaded into the CU.

The second sw part are applications, e.g. **setr**.exe used for the service communication between the PC and CU. Others applications are e.g. **memcp**.exe or **netcnf**.exe.

Firmware modules in MR400 memory:

This also applies for MR900, MR300, MR160, MC100, MG100.

The CU contains two memory areas in S-RAM memory used for firmware modules:

1. **B** - Basic module

   is the basic module which is started after resetting the CU, subsequently the program switches to module E.

2. **E** or another version D, G, H, I...

   is the main programming module which can be replaced by modules D or G, H, I... These modules primarily differ in the user protocols used.

Firmware modules in MR25 memory:

This also applies for MR25ET, MCM302.

The CU contains four memory areas in Flash memory used for firmware modules:

1. **B** (B-saver) - Basic module

   is the basic module which is started after resetting the CU, subsequently the program switches to module E.

2. **E** or others - main module, can exist in next versions E, D, G, H, I...

   is the main programming module which can be replaced by modules D or G, H, I... These modules primarily differ in the user protocols used. The main module allows the loading modules A or W or B.

3. **A** or **W** - Air or Wire module

   A module is used in the case of downloading the main module (E) through the radio channel. The A module can be replaced by module W for downloading the main module over the wire link between the SCC of two CU.

4. **C** (C-loader)

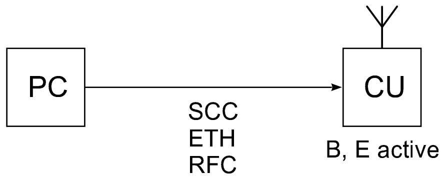
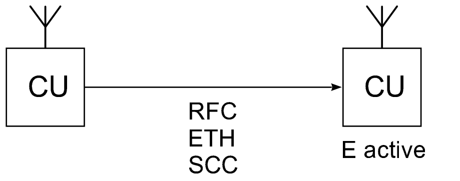
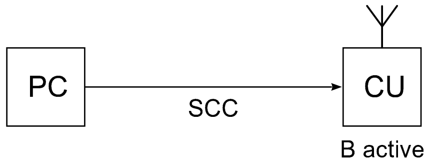
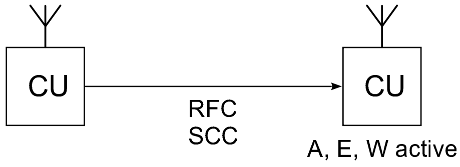
   is used in the MR25ET and MCM302 for downloading via D-RAM.

The **firmware versions** are numbered as they are developed, e.g. version 7.45 was created in January 2006. Using the command `(s)ervice (v)ersion` in Setr we obtain a list of present modules in CU e.g.:

```
D745
B745
```

It is not necessary to have equal numbers in all modules, but for the possible use of older modules you should consult with the sw section in Racom. The latest version of the A or W module is now (10/2006) the A602 or W602.

There are 4 situations when loading fw into CU:

	memcp	setr+itl
MR400		
MR25		

Firmware can be downloaded to a CU in two ways:

MR400 via D-RAM

This method is used in MR900, MR400, MR300, MR160, MC100, MG100, MR25ET, MCM302, which contain auxiliary D-RAM. The CU runs in module E and transfers data which the new module E stores in D-RAM. S-RAM remains unchanged and the CU communicates normally with other members of the network. Only after transmission of all data and its inspection is the module copied within a few seconds from the auxiliary D-RAM to S-RAM. From the point of view of interference this method is safer than the following method.

MR25 directly to Flash memory

Transferred data is written directly to Flash memory. Meanwhile the CU is operating in a different module than the one which is downloaded. For example, the CU is running in module A and module E is downloaded. During downloading (several minutes) the CU can perform routing, however, none of its other functions. It is used in MR25, MR25ET, MCM302.

It is possible to use 2 types of software from the MORSE package for downloading:

memcp download **from PC to CU** over the service lead, Ethernet or network - program **MEMCP**, see article Chapter 3, *Downloading firmware using Memcp*

setr+itl download **from CU to another CU** over the MORSE network - program **SETR** and its service **itl**, see article Chapter 4, *Downloading using Setr and itl*

An overview of source files for download[1], which are mentioned in other chapters. Only examples from individual groups are mentioned.

MR25	MR400
Files for download from www.racom.eu	
m1r-745-me.zip m1r-745-m-all.zip mr25-602-ma10.zip mr25-602-mw10.zip	m2r-745-ce.zip m2r-745-c-all.zip
Files containing fw in the directory e.g. c:\MORSE-745\fkl\	
me10.fkl	ce10.fkl
ma10.rfr	
mw10.rfr	
Batch files in c:\MORSE-745\	
mr25_cbl_le.bat	mr900_cbl_le.bat
Works also for:	
MR25ET, MCM302	MR900, MR300, MR160, MC100, MG100

⚠ **Important**

After fw download it is necessary to save the initial configuration into FLASH memory using Setr and command `(c)nf (b)ackup`.

[1] https://www.racom.eu/cz/download/

2. Simple Downloading using MEMCP

The macro mr900_cbl_le.bat contained in the sw package, for example `mswin-i386-745-utils.zip`, is suitable for saving module E from the computer to the CU **MR400** using the service cable.

For it to function it is necessary for the directory, where the macro `mr900_cbl_le.bat` is located, to contain the file `memcp.exe`, the auxiliary program `starter.exe` and the subdirectory `fkl`, containing the file `ce10.fkl`. If the setr.exe program is running, switch it off.

Start the macro

```
mr900_cbl_le.bat
```

wait about 10 sec until a communication link is made and monitor the transfer of packets indicated by symbols ###. After transfer is complete switch off the download message using the **Alt+F4** command and the module is ready for use in the CU.

Other macros containing mr900_cbl_... are used in the same way:

```
mr900_cbl_le.bat
mr900_cbl_ld.bat
mr900_cbl_lg.bat
mr900_cbl_lb.bat
```

Below is a general description of the macro. For a more detailed definition see the article Chapter 3, *Downloading firmware using Memcp*. Macros for downloading individual modules:

```
starter memcp -nt5000 -nr20 -as8 -pS -af.\fkl\ce10.fkl -pb115200 %1 %2 %3
starter memcp -nt5000 -nr20 -as8 -pS -af.\fkl\cd10.fkl -pb115200 %1 %2 %3
starter memcp -nt5000 -nr20 -as8 -pS -af.\fkl\cg10.fkl -pb115200 %1 %2 %3
starter memcp -nt5000 -nr20 -as4 -pS -af.\fkl\cb10.fkl -pb19200 %1 %2 %3
```

starter	program starter calls step-by-step the applications written in the batch file
memcp	calling the memcp program
-nt5000 -nr20	timeout and repeat for transmission through the network
-as8	length of transmitted packets
-pS	B-saver module active in the destination CU
-pb115200	signalling rate over service cable
-af.\fkl\ce10.fkl	calling source file `ce10.fkl`
%1 %2 %3	place for other possible parameters

When downloading various modules only the labelling of the source file `ce10.fkl` changes. For increased security when transmitting module B the length of the packet is shortened to `-as4` and the signalling rate is reduced to `-pb19200`.

3. Downloading firmware using Memcp

Detailed description, valid from version 616.

Recommended organisation of directories which suits batch files from version 674:

```
morse
   fkl
      cb10.fkl
      ce10.fkl
      ...
   memcp  .exe
   setr   .exe
   starter.exe
   mr900_cbl_le.bat
   ...
```

`*.fkl` files are used for downloading. Only modules `ma10`, `mw10` are in version `*.rfr`.

3.1. Downloading from the PC to MR400 (memcp)

- Source = PC
- Destination = MR400, MR900, MR300, MR160, MC100, MG100
- Modules D, E, G, H,...., B

Use some of the prepared batch files and where necessary add other parameters. The required CU configuration is similar to the configuration required for sending a "!" command to the respective CU.

- **locally over the service lead**
  - use the batch file

```
mr900_cbl_le.bat
```

which starts utility memcp.exe with parameters

```
starter memcp -nt5000 -nr20 -as8 -pS -af.\fkl\ce10.fkl -pb115200 %1 %2
```

- **via the local CU and through the MORSE network** to the CU with address `690F5502`

  - use batch file (net = network) and add two parameters:

```
mr900_net_le.bat -pb115200 -da690F5502
```

which starts utility memcp.exe with parameters

```
starter memcp -nt5000 -nr20 -as8 -af.\fkl\ce10.fkl %1 %2 %3
-pb115200 -da690F5502
```

- **locally via the Ethernet connection**
  - use batch file and add the IP address of ETH channel in CU

```
mr900_net_le.bat -pIP192.168.0.9
```

which starts utility memcp.exe with parameters

```
starter memcp -nt5000 -nr20 -as8 -af.\fkl\ce10.fkl %1 %2 %3
-pIP192.168.0.68
```

- **by Ethernet via the locally CU and next through the MORSE network** to `690F5503`
  - use batch file and add the IP address of ETH channel in the CU and the remote MORSE address. The MAS configuration is necessary.

```
mr900_net_le.bat -pIP192.168.0.9 -da690F5503
```

which starts utility memcp.exe with parameters

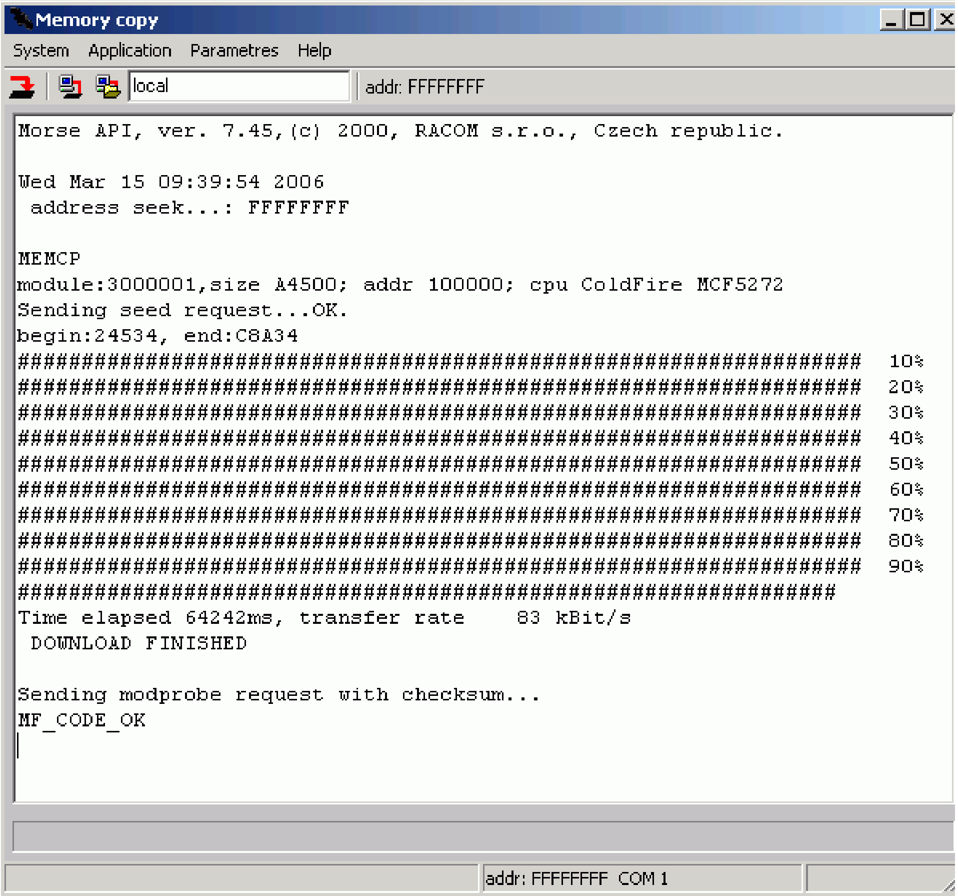
```
starter memcp -nt5000 -nr20 -as8 -af.\fkl\ce10.fkl %1 %2 %3
-pIP192.168.0.68 -da690F5503
```

Meaning of the utilities and parameters used

starter	program starter ensures that Windows applications contained in `file.bat` are called in succession if there are more of them than in `mr25_net_le_et.bat`
memcp	calling program memcp, which performs transmission of the fw module by individual packets
-nt5000 -nr20	timeout and repeat for transmission through the network
-as8	the number of 128 byte sectors transmitted in a single packet. If connection conditions are poor select a small number, e.g. x=2, and use x=8 in the case of very good conditions and for transmission over the service lead
-pS	The B-saver module is active in the destination CU. If the parameter is left out then the main module is in operation during transmission, e.g. E (in MR25 this is only possible for transmission of modules B, A, W).
-pb115200	the transfer speed over the service lead. The parameter activates transmission via SCC
-pIP192.168.0.9	IP address of the ETH channel of the CU to which the PC is connected via Ethernet. The parameter activates transmission via ETH
-af.\fkl\ce10.fkl	calling the source file ce10.fkl. When downloading module B `cb10.fkl` particular care should be taken. If a fault occurs on downloading module B it is necessary to send the CU to Racom to have new sw installed. In most cases it is not necessary to download a new module B, however, always check the number of its version before downloading software to the real radio network.

%1 %2 %3 location for possible additional parameters

-da690F5502 address of the destination CU; this is used if a locally connected CU is only an intermediary for sending fw packets over the MORSE network to another CU. This is conditional upon routing from the PC to the destination CU being functional. Test using an "exclamation mark" command



```
Morse API, ver. 7.45,(c) 2000, RACOM s.r.o., Czech republic.

Wed Mar 15 09:39:54 2006
 address seek...: FFFFFFFF

MEMCP
module:3000001,size A4500; addr 100000; cpu ColdFire MCF5272
Sending seed request...OK.
begin:24534, end:C8A34
###################################################################  10%
###################################################################  20%
###################################################################  30%
###################################################################  40%
###################################################################  50%
###################################################################  60%
###################################################################  70%
###################################################################  80%
###################################################################  90%
#################################################################
Time elapsed 64242ms, transfer rate    83 kBit/s
 DOWNLOAD FINISHED

Sending modprobe request with checksum...
MF_CODE_OK
```

Fig. 3.1: Memcp after finishing the download

Downloading `ce10.fkl` over the service lead takes about 60 sec, via Ethernet about 6sec, followed by 6 sec to overwrite transferred data from D-RAM to S-RAM, and finally a reset.

3.2. Downloading from the PC to MR25 (memcp + B-saver)

- Source = PC
- Destination = MR25, MR25ET, MCM302
- Modules D, E, G, H,...., B, A, W

MR25 has a different memory configuration, which is why different procedures are used than for MR400.

Here we use a pair of parameters **-ar -aE**, which causes a direct entry to flash memory.

Parameter **-pS** switches the station to module B, which can transmit all other modules, however, only locally.

By ignoring -pS the station remains in the main module (E), it can only transmit modules B,A,W. This transmission with parameter -da is also possible through the MORSE network to another CU.

Use of modules A,W is described in chapter Download using itl.

- **locally over the service lead**
  - use the batch file

  ```
  mr25_cbl_le.bat
  ```

  which starts utility memcp.exe with parameters

  ```
  starter memcp -nt5000 -nr20 -ar -aE -as8 -pS -af.\fkl\me10.fkl -pb115200 %1 %2
  ```

  for transmission of module B use the batch file

  ```
  mr25_cbl_lb.bat
  ```

- **via the local CU and through the MORSE network** to the CU with address `690F5502`, only modules **B, A, W**
  During transmission of module B, A, W the main module is in operation (-pS missing), the local version or remote version with parameter `-da` is therefore possible

  - use batch file (net = network) and add two parameters:

  ```
  mr25_net_lb.bat -pb19200 -da690F5502
  ```

  which starts utility memcp.exe with parameters

  ```
  starter memcp -nt5000 -nr20 -as8 -af.\fkl\mb10.fkl %1 %2 %3
  -pb19200 -da690F5502
  ```

The meaning of parameters is the same as in the previous paragraph.

When downloading `ma10.rfr` or `mw10.rfr` there are the same possibilities as in the case when downloading `mb10.fkl`. We can use `-pS`, and then download locally using B-saver or omit parameter `-pS`, and then work using the main module with the possibility of using remote access:

```
 memcp -af.\fkl\ma10.rfr -as8 -pb115200 -ar -aE -pS             or
 memcp -af.\fkl\ma10.rfr -as8 -pb115200 -ar -aE     -da690F5502
```

Batch files:

In the sw package, e.g. `mswin-i486-625-morse.zip` there is a set of files prepared for downloading. Here we present the names and content of files determined for working with the B-saver in MR25:

```
mr25_cbl_le.bat
mr25_cbl_ld.bat
```

```
   mr25_cbl_lg.bat
   mr25_cbl_lh.bat
   mr25_cbl_la.bat
   mr25_cbl_lb.bat
   mr25_cbl_lb_up.bat
```

```
memcp.exe -nt5000 -nr20 -ar -aE -as8 -pS -af.\fkl\me10.fkl -pb115200 %1 %2 %3
memcp.exe -nt5000 -nr20 -ar -aE -as8 -pS -af.\fkl\md10.fkl -pb115200 %1 %2 %3
memcp.exe -nt5000 -nr20 -ar -aE -as8 -pS -af.\fkl\mg10.fkl -pb115200 %1 %2 %3
memcp.exe -nt5000 -nr20 -ar -aE -as8 -pS -af.\fkl\mh10.fkl -pb115200 %1 %2 %3
memcp.exe -nt5000 -nr20 -ar -aE -as8 -pS -af.\fkl\ma10.rfr -pb115200 %1 %2 %3
memcp.exe -nt5000 -nr20 -ar -aE -as4 -pS -af.\fkl\mb10.fkl -pb19200  %1 %2 %3
memcp.exe -nt5000 -nr20 -ar -aE -as8     -af.\fkl\mb10.fkl -pb19200  %1 %2 %3
```

We see that the batches contain parameters `-ar -aE` for direct entry to flash memory and parameter `-pS`, which activates module B. Transmission runs over the service lead. In the last batch `-pS` is omitted, which is why module E will be in operation (used if the present module B was older than sw616).

In the macro it is possible to change a parameter by editing or add a parameter to the command line. After transferring the name of the macro to the command line (Ctrl + Enter) it is possible to add the missing parameter. If the parameter is already contained in the macro then the connected parameter replaces its value. For example:

```
mr25_cbl_le.bat -as4
```

will transmit module E using shorter packets.

Downloading from the PC to MR25ET and MCM302 (memcp + loader)

These CU use `me10.fkl` series modules such as MR25, however, they come with D-RAM and therefore in addition they permit transmission over the network. The above mentioned procedures for MR25 for the transmission of modules C, D, E, G, H,... are growing in number. They are used in a similar way as in the case of MR400, in place of batch file `mr900_net_le.bat` we `use mr25_net_le_et.bat`.

For the purpose of downloading module C-loader supplements modules B, A, W, E, D, G, H in MR25ET. Module B-saver has been modified from version 616, the old module B needs to be updated with the new one using `mr25_cbl_lb_up.bat`.

Downloading is done in two steps. First download the short C-loader, and with the next command download the main module using this loader.

- **locally over the service lead**
  the batch file with parameter

  ```
  mr25_net_le_et.bat -pb115200
  ```

  starts the memcp.exe utility with parameters

  ```
      memcp -af.\fkl\mc10.fkl -as8 -pb115200
      memcp -af.\fkl\me10.fkl -as8 -pb115200
  ```

- **via the local CU and through the MORSE network** to the CU with address 690F5502

```
mr25_net_le_et.bat -pb115200 -da690F5502
```

```
    memcp -af.\fkl\mc10.fkl -as8 -pb115200 -da690F5502
    memcp -af.\fkl\me10.fkl -as8 -pb115200 -da690F5502
```

- **by Ethernet**

```
mr25_net_le_et.bat -pIP192.168.0.9
```

```
    memcp -af.\fkl\mc10.fkl -as8 -pIP192.168.0.9
    memcp -af.\fkl\me10.fkl -as8 -pIP192.168.0.9
```

- **by Ethernet via the locally CU and next through the MORSE network** to 690F5503:

```
mr25_net_le_et.bat -pIP192.168.0.9 -da690F5503
```

```
    memcp -af.\fkl\mc10.fkl -as8 -pIP192.168.0.9 -da690F5503
    memcp -af.\fkl\me10.fkl -as8 -pIP192.168.0.9 -da690F5503
```

Downloading of the C-loader takes approx. 2sec, downloading of `me10` takes about 45 sec over the service lead, via Ethernet about 6sec, followed by 6 sec to overwrite the transferred data from the D-RAM to Flash, and finally reset.

Note

Replacing new sw with old can be more difficult. The key to it is downloading the old (519) module B using the new (640) memcp.exe and then using the old (585) swl.exe download the old main module (E 585).

Other notes on firmware options are given in chapter Chapter 5, *Notes*

4. Downloading using Setr and itl

The downloading of firmware over the radio channel is a relatively complicated task containing various dangers for the network. This is why it is only recommended for experienced operators.

The downloading mentioned here allows for the transfer of firmware modules between Communication units (CU) in the MORSE network. It uses the **itl** service of the SETR program, which allows both ways of loading the firmware module into the CU:

- via D-RAM memory for MR400, see article 1. and 2.
- directly into Flash memory for MR25, see article 3. and 4.

4.1. Downloading the main module E using the kernel module into MR400

These conditions must be accomplished:

- good connection between the source and destination CU via some MORSE channel (radio, ethernet, SCC), in case of the radiochannel a direct connection without retranslation is recommended
- the destination CU contains the D-RAM, so it is the MR400, MR160, MR900, MC100 (but not MR25)
- the source CU contains this main module (E or D or ...), which should be transfered in the destination CU

In program **Setr.exe** the menu **memload** is filed amongst tests. Open using command **itl** and fill in the following items:

(N):1 choice of the node for communication with the destination CU

(d):690F0002 address of destination CU

(x):4 number of 128 byte sectors transferred in one packet, for wrong conditions of the radio connection choose the lower number, e.g. x=2, for very good conditions use x=8

(m):m module mode

(y):k module type kernel, the choice is indicated by the address pair 01000001, 03000001 on the next row

```
Memload:
(N):1   (d):690F0002h
(E)xternal flash:OFF
ma(x) sectors:4   (m)odule:MODULE   preset t(y)pe
User module: fi(r)st:01000001   (l)ast:03000001
(t)imeout:12000
(s)tart r(e)port sto(p)
go MORSE (A)/(W)
(i)nit
```

```
(f)ire (k)ill (c)ontinue
check through (S)UM32
go MORSE (B)
(C)..modprobe with chksum
(M)odprobe
(q)uit
>>
```

Continue with next commands:

(s) Enter start test itl

(i) Enter init in the destination CU

(e) Enter report - check the test report before work:

```
This is Memload v1.03 response
max. MF sectors per packet :4
loading module:  -
target        : 690F0002h
status :ready; Check result: none
begin  :FF040000h
end    :FF0FA580h
current:FF040000h
talking to MORSE E
timeout 12000
Time elapsed: 0msec
Transfer rate: -1.#IND00kbps
>>
```

The transfer starts using **(f)ire** and follow the progress using **r(e)port**:

(f) Enter start transmission, LED indicate intensive operation on the RF channel

(e) Enter report about the transfer progress:

```
...
status :waiting for memfill response; Check result: none
begin  :FF040000h
end    :FF0FA580h
current:FF053600h
...
```

The address `current` increases from address `begin` to the address `end`. If necessary, we can interrupt the process using `(k)ill` and go on using `(c)ontinue`. Transmission of the main module takes 10 to 15 minutes depending on network conditions at the time.

There is no special message indicating the end of transmission. However, we recognise that transmission has ended when the LED for RF transmission and receipt stops flashing. Definite information is contained in the reply to a `r(e)port` query, where address `current` has reached or exceeded address `end`:

```
status :all sectors are transfered.; Check result: none
begin  :FF040000h
end    :FF0FA580h
current:FF0FA580h
```

(C) Enter the transferred kernel in the destination CU is checked and copied from D-RAM to the permanent memory S-RAM. It takes about 6 seconds.

(e) Enter report:

```
status :pal O.K.; Check result: none
```

(p) Enter stop test `itl` in the source CU

Downloading of the firmware module is completed.

The summary of commands used for MR400, main module, follows:

```
itl
  N1 d690F0002 x4
  mm      ... module mode
  yk      ... module kernel
  s i e   ... test start
  f e     ... loading
  C e     ... copy to S-RAM
  p       ... stop test
```

Module B-saver can be transferred similarly, like kernel module:

```
itl
  N1 d690F0002 x4
  mm      ... module mode
  ys      ... module saver ... different choice from kernel
  s i e   ... test start
  f e     ... loading
  C e     ... copy to S-RAM
  p       ... stop test
```

When transferring B-saver we receive a different message in the identification of modules 01000004, 03000004:

```
Memload:
(N):1   (d):690F0002h
(E)xternal flash:OFF
ma(x) sectors:4   (m)odule:MODULE   preset t(y)pe
User module: fi(r)st:01000004   (l)ast:03000004
```

The range of transferred addresses is smaler:

```
begin  :FF030000h
end    :FF037400h
current:FF030000h
```

Transmission of the B-saver module takes about 30 sec.

4.2. Downloading the main module E using the loader and kernel module into MR25ET

These conditions must be accomplished:

- good connection between the source and destination CU
- the destination CU contains the D-RAM, so it is the MR25ET or MCM302 (but not MR25)
- the source CU contains this main module (E or D or ...), which should be transfered in the destination CU
- the source CU contains the loader module `mc10.fkl`, best newly loaded using the command `memcp -af./fkl/mc10.fkl -as8 -pb115200`

The procedure is the same as in the previous case. The difference is that first we transfer a small module loader, select **(y):l**,and then we repeat the whole process for the main module **(y):k**. The module loader is identified by letter C, for example C757 is in the reply to an `sv` command. An overview of Setr commands used is given below :

```
itl
  N1 d690F0002 x4
  mm      ... module mode

  yl      ... module loader
  s i e   ... test start
  f e     ... loading
  C e     ... copy to S-RAM
  p       ... stop test

  yk      ... module kernel
  s i e   ... test start
  f e     ... loading
  C e     ... copy to S-RAM
  p       ... stop test
```

Module B-saver can be possibly transferred also:

```
  ys      ... module saver
  s i e   ... test start
  f e     ... loading
  C e     ... copy to S-RAM
  p       ... stop test
```

4.3. Downloading of module A or W or B through module E into MR25

These conditions must be accomplished:

- good connection between the source and destination CU
- the destination CU is MR25
- the destination CU is MR25ET or MCM302 - the alternative method to the above mentioned one
- the source CU contains this module (A or W), which should be transferred to the destination CU

The procedure is the same as in the previous cases, differences are outlined below:

itl	memload test in Setr
(d):690F8100	address of destination CU
(x):4	number of 128 byte sectors transferred in one packet
(m):A	module transferred, A represents both A or W
(s) Enter	start test itl
(i) Enter	init in the destination CU
(e) Enter	report - check the test report before work:

```
This is Memload v1.03 response
max. MF sectors per packet :4
loading module: MORSE A
target        : 690F8100h
status :ready Check result: none
begin  :  374000h
end    :  37D800h
current:  374000h
talking to MORSE E
timeout 12000
Time elapsed: 0msec
Transfer rate: nankbps
```

where is:

```
 loading module: MORSE A - modul A(= A or W) will be transferred
 talking to MORSE E      - the transfer will be done using module E (E or D or ...)
```

(f) Enter	start transmission
(e) Enter	report about the transfer progress

When completed, the report looks like this:

```
status :all sectors are transfered. Check result: none
begin  :  374000h
end    :  37D800h
current:  37D800h
```

(S) Enter a checksum of the transferred module

(e) Enter the previous message appears, however containing the row:

```
status :all sectors are transfered. Check result: O.K.
```

In the case, when the `Check result` is not `O.K.`, stop the test and start again using the command `(s)tart`, see previous article.

(p) Enter stop test `itl` in the source CU

Downloading of the firmware module is completed.

The summary of used commands follows:

```
itl
  N1 d690F8100 x4
  mA       ... module transferred
  s i e    ... test start
  f e      ... loading
  C e      ... copy to S-RAM
  p        ... stop test
```

4.4. Downloading of module E or B through module A or W into MR25

These conditions must be accomplished:

- good connection between the source and destination CU
- the destination CU is MR25 or
  the destination CU is MR25ET or MCM302 - the alternative method to the above mentioned one
- the source CU contains this module (D, E...), which should be transferred to the destination CU
- the destination CU contains module A, if we are going to transfer the main module on air, or
  the destination CU contains module W, if we are going to transfer the main module via wire link

Procedure:

itl	memload test in Setr
(d):690F8100	address of destination CU
(x):4	number of 128 byte sectors transferred in one packet
(m):E	module transferred, E represents the current main module in the source CU (E, D, G, H, I), B is the base module (B-saver)
(s) Enter	start test itl
(i) Enter	init in the destination CU
(A) Enter	switch the dest. CU in the module A

or

(W) Enter	switch the dest. CU in the module W in case of downloading via wire
(i) Enter	init in the destination CU
(e) Enter	report - check the test report before work

Sometimes it happens that a similar message appears:

```
status :ready Check result: none
begin  :  308000h
end    :  36A400h
current:  308000h
                   E via E???
talking to MORSE E
```

In this case switching from E to A module in the destination modem did not happen and we will repeat the commands:

(A) Enter switch the dest. CU in the module A

(i) Enter init in the destination CU

(e) Enter check report

The right answer looks like this:

```
status :ready Check result: none
begin  :  308000h
end    :  36A400h
current:  308000h
talking to MORSE A(W)
```

where:

```
  talking to MORSE A(W)   - the transfer will be done using module A or W
```

(f) Enter start transmission

(e) Enter report about the transfer progress

When completed, the report looks like this:

```
status :all sectors are transfered. Check result: none
```

(S) Enter a checksum of the transferred module

(e) Enter the message contains the row:

```
status :all sectors are transfered. Check result: O.K.
```

The destination CU runs in module A or W. In this state (when downloading and after it has finished) the CU is able to perform routing and only certain commands. The short reply **A602** follows an **sv** query and after query:

(N)ode (e)dit Enter the message appears:

```
690F8100h> @  A/W: Service not available!
```

So after downloading has finished it is necessary to switch the destination CU in the main module again:

(B) Enter switching the destination CU in B module

(i) Enter restart of the destination CU is executed, the CU later goes into the main module

(e) Enter checking if the destination CU is switched back into E module:

```
This is Memload v1.03 response
max. MF sectors per packet :4
loading module: MORSE E
```

```
target        : 690F8100h
status :ready; Check result: none
begin  :  308000h
end    :  36A400h
current:  308000h
                         E via E???
talking to MORSE E
timeout 12000
Time elapsed: 901158msec
Transfer rate: 0.000000kbps
```

If this message appears, then the destination CU is O.K. and we can stop the test:

(p) Enter stop the memload test in the source CU

Downloading of the main module is finished.

The summary of commands used follows:

```
itl
  N1 d690F8100 x4
  mE      ... module transferred
  s i     ... test start
  A i e   ... switching dest CU in A module
  f e     ... loading
  S e     ... check sum
  B i e   ... switching through B into E
  p       ... stop test
```

If the basic module B is transferred in this way it is important not to interrupt operation on the destination CU until transmission is correctly finished. This means not restarting, switching off the power supply or using command **go MORSE (B)**. After restarting the CU module B is started and if this module has not been fully transferred then the program doesn't work. It is then necessary to complete or retransmit B whilst the CU is still in the auxiliary module A or W.

Note

All operations can be done using remote access. In this way it is possible to download the firmware module step by step in the whole network.

When the main module is downloaded via module A, then the destination CU can fulfil the others functions in limited range only, until it is rebooted again. The CU running in the A module can do store and forward relaying in the network, but in spite of this, it is recommendable to reduce the traffic in a heavily loaded network when downloading.

The `(i)nit` command in the "itl" can be used more often when we are in doubt about good progress of work, e.g. when switching to the A module.

The versions A or W older than 602 in one CU do not work with new (602 and later) sw in the second CU. It is recommended to replace them by version A602 or W602 or later, if available.

The modules `ma10.rfr` and `mw10.rfr` are contained in the firmware version 602.

5. Notes

Versions of fw and sw

It is recommended to use the same version of firmware in CUs and sw in PCs. For example, firmware modules E757 and B757 in CUs and Setr from the morse-757 package in PCs.

Other combinations are possible, however, we may come across exceptions here brought about by the continuous development of MORSE software. The most important of them are given below. For example, module B was modified when switching from version 6xx to 7xx. Therefore it is also necessary to replace module B685 with module B705 when replacing module E685 with module E705. On the other hand between versions 730 and 745 it is not necessary to change module B. It is better to avoid unnecessarily downloading module B, particularly during transmission over the radio channel as this presents a certain risk to the CU.

When replacing old fw with new it is recommended to first download the new module B and then the new main module. When replacing the main module with a new one of the same version, e.g. replacing E730 with D730 then module B remains unchanged.

Configuration

If during development of sw a new parameter is added to a certain menu, then the configuration structure of the new version of firmware shall change. After downloading there may be a loss of configuration parameters and so it will be necessary to add them manually. It is therefore recommended to back-up the finished configuration using command `(c)nf (b)ackup`, i.e. **cb**. The backed-up configuration is then used automatically on restarting or manually on entering command **ce**.

A change in configuration structures also causes a message to appear when trying to enter `write` or `Init`, for example using Setr 745 to CU with fw 757:

```
get NODE 0 Invalid cnf version! Obsolete client!
```

The old Setr can operate with new fw in the CU, however, it cannot write. We can correct this by using Setr 757 or a later version.

Changes

Be careful when downloading the older main module (<602) instead of the new one (>=602). After this it is necessary to return the parameters to the default state (e.g. using **ca** and **sgB** commands) and set the parameters again manually.

Module B-saver is rewritten from version 616, the old module B needs to be replaced with this new one. Don´t use sw600 to sw615 for downloading the old B < sw600

The module B-saver is rewritten from version 616. The old module B must be replaced by the new one.

If we download new sw (>sw600) to the CU, it is then necessary to check whether module B, version 616 or higher, is in the destination CU, and if not then to download it: `memcp -af.\fkl\mb10.fkl -ar -aE` We can then download other modules using this new B-saver.

Other significant changes were made from B 641. We should download them in order: firstly kernel (e.g. E 641), then B 641.

When changing from version 6xx to 7xx it is also necessary to download module B. Use batch file `mr900_cbl_lb_up.bat`, which overwrites module B using the main module.

For version fw 757 and higher it is also necessary to use Setr 757 and higher.