...the broadest narrowband money can buy

RACOM
RADIO DATA NETWORKS

SETR software for MR25

MORSE Firmware Documentation

version 1.17
2/3/2010

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. MORSE main menu [sv 7.26] 5
    1.1. Managing the MORSE main menu 5
    1.2. CU Configuration Services 8
    1.3. Useful tips 9
2. Hardware [sv 8.03] 11
3. Communication Unit [sv 7.26] 12
4. Radio part [sv 9.30] 15
    4.1. Radio hardware 15
    4.2. Radio parameters 18
    4.3. Radio calibration 19
5. Nodes [sv 8.03] 21
6. Morse Redundancy Solution [sv 9.30] 24
    6.1. Node extensions 24
    6.2. Globals - Mobile mode 25
    6.3. Hierarchy routing table 31
7. Broadcasting [sv 7.26] 32
8. SC Channels [sv 7.26] 34
    8.1. SC Channel to Node Interface 34
    8.2. SC Protocol 34
    8.3. SC eXtensions 37
9. RF Channels [sv 10.0.28.0] 38
    9.1. RF Channel to Node Interface 38
    9.2. RF Access 38
    9.3. RF Protocol 40
    9.4. RF Modem 43
10. Ethernet [sv 10.0.94.0] 45
    10.1. Ethernet Channel to Node Interface 45
    10.2. Hardware configuration 45
    10.3. Setting protocol parameters 47
11. NAG Channels [sv 7.26] 60
    11.1. NAG Channel to Node Interface 60
    11.2. NAG Protocol 60
12. Address resolution table [sv 7.26] 62
13. Routing (retranslation) tables [sv 7.26] 65
14. Menu (c)nf [sv 7.45] 70
15. Misc - supplementary functions [sv 7.26] 71
16. Service [sv 7.26] 72
    16.1. (h)eap info 72
    16.2. (V) (v)ersion 72
    16.3. RF service (R)X 73
    16.4. RFservice (T)X 74
    16.5. (e)cho 74
    16.6. unit (s)tatus 74
    16.7. (t)ime 76
    16.8. (g)o 77
    16.9. (c)lock info 77
    16.10. loop (i)nfo 77
    16.11. (r)adio mem c(o)mp ch(k) 77
    16.12. (m)obil 78
17. Diagnostic [sv 7.26] 79
18. Monitoring 80

18.1. Routing of the monitoring messages 81
18.2. Monitoring SCC channels 83
18.3. Monitoring RFC channels 87
18.4. Monitoring ETH channels 93
18.5. Monitoring NAG channels 98
19. Statistic logs 99
19.1. LINKS LIST 101
19.2. CALLS LIST 103
19.3. SCC PORTS LOG 104
19.4. NET USER LOG 104
19.5. RF MEDIUM LOG 105
19.6. ETHERNET LOG 106
20. Events 108
21. Tests 110
21.1. STATISTIC TEST 110
21.2. BER TEST 114
21.3. SEND PACKET 115
21.4. PING TEST 116
21.5. ROUND TEST 117
21.6. MEMLOAD 119
22. Channel send 123
22.1. Enable RF(T)X 124
23. Path editor [sv 7.26] 125
23.1. Path packets and communication with remote stations 126
24. Help [sv 7.26] 128
25. CNI-channel to node interface [sv 9.30] 129
Index 133
A. Revision History 135

List of Figures

1.1. Radiomodem MR25 5
1.2. Before making command **edit** (i.e. in menu, which contains command edit 9
1.3. After making command **edit** (menu doesn't contain command edit) 9
4.1. Before edit execution (i.e. in the menu, which contains the command edit). 15
4.2. After the edit function. 15
8.1. Signal way through interface 35
10.1. ARP table 50

1. MORSE main menu

1.1. Managing the MORSE main menu

This description is valid for the communication units MR25, MR25ET and MCM302.

Immediately after starting, the Setr.exe application tries to establish connection to a MORSE node. Then the MORSE main menu appears.

Fig. 1.1: Radiomodem MR25

There is a MR25 unit. Connector on the panel is used for the service access by a terminal software. The connector wiring is described in MR25 Service Manual, see Support, Radio modems, MR25, Manual.

MORSE main menu

```
MORSE main menu:

(H)W   (U)nit (R)adio
(N)odes  s(D)r   (B)c
(S)CC   R(F)C   (E)th
(A)rt   r(T)ab  (c)nf
(m)isc  NA(G)

(s)ervice      d(i)ag
(p)ath        (?)help

(o)ld cnf menu

(q)uit
```

Meaning of menu components

(H)W	HW data (written during production)
(U)nit	general parameters applicable to whole CU
(R)adio	parameters of the radio part
(N)odes	node parameters

s(D)r	semidynamic routing, in development
(B)c	broadcast routing for all nodes
(S)CC	serial communication channels
R(F)C	radio frequency channels
(E)TH	Ethernet channel
(A)rt	address resolution tables
r(T)ab	routing tables (store and forward retransmission)
(c)nf	general configuration services
(m)isc	miscellaneous services for special cases
NA(G)	network agent, commun. channel without physical output
(s)ervice	unit diagnostic tools
d(i)ag	network diagnostic tools (logs, tests, monitoring)
(p)ath	path packet editor
(?)help	help for shell commands
(o)ld cnf	menus for obsolete services and configurations (for units with version 3.52 and older)

Each of the letters in parentheses represents one submenu. To choose one of them, type the respective letter and press the **Enter** key. When the submenu appears, carry out the next choice. It is necessary to distinguish between small and capital letters.

It is possible to return to the main menu from any level of the menu using the escape key, or **Q Enter**. By selecting option **q Enter**, which is contained in each menu, you step back by one menu level.

It is possible to return to the main menu from any level of the menu using the escape key, or Q Enter. The choice q Enter, which is contained in each menu, accomplish one menu level step back.

Parameters are entered by typing the appropriate letter, immediately followed by the desired parameter value. If the parameter is entered with only a letter, a short prompter to this edited parameter appears.

A sample example:

In the Main menu choose **U Edit**, thus the submenu `(U)nit`

```
Communication unit:
(e)dit
rmt de(f)ault  rmt (r)ead  rmt (w)rite
(q)uit
>>
```

Next **e Enter**

```
get  UNIT 0 O.K.
>>
```

The get operation was successfully finished – the set of unit configuration parameters has been received by Setr.exe and stored in a temporary buffer. Press **Enter** (To see parameters values)

```
Communication unit:
service (d)est:00000000 (N):0
(U)cc limit:130 * 0.1V  (h)yst.:2 * 0.1V
(s)tatus:0000 (P)SWD:OFF (L)OGW:OFF (M)INP:OFF
(l)og period:86400
Time (Z)one for DST:EU Time zone DI(F)F:3600sec
            obsolete(<605) Time (z)one:69136sec
Summer time (all to zero - off):
start (1)month:3 (2)day:27
end   (3)month:10 (4)day:30
Time sync:
 (i)n:0 Sr(c):00000000 (p)eriod:0
(O)ut:0 ds(t):00000000 p(e)riod:0
P(A) dir:0000 dat(a):0000
P(B) dir:0000 dat(b):0000
Power saving:
alar(m) time:0s m(o)de:0000
(R)adio pll recovery:OFF
de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
>>
```

For example the item `DIF(F):3600sec` shows how many seconds the unit must add to GMT to get local time. If you need to change this parameter from 1 hour to 2 hours, type: **F Enter**. A short prompt appears:

```
(q)uit
write new seconds to get localtime from GMT:
```

Write new count of seconds **7200 Enter** and add **Enter** for displaying:

```
Communication unit:
service (d)est:00000000 (N):0
(U)cc limit:130 * 0.1V  (h)yst.:2 * 0.1V
(s)tatus:0000 (P)SWD:OFF (L)OGW:OFF (M)INP:OFF
(l)og period:86400
Time (Z)one for DST:EU Time zone DI(F)F:7200sec
            obsolete(<605) Time (z)one:72736sec
Summer time (all to zero - off):
start (1)month:3 (2)day:27
end   (3)month:10 (4)day:30
Time sync:
 (i)n:0 Sr(c):00000000 (p)eriod:0
(O)ut:0 ds(t):00000000 p(e)riod:0
```

```
P(A) dir:0000 dat(a):0000
P(B) dir:0000 dat(b):0000
Power saving:
alar(m) time:0s m(o)de:0000
(R)adio pll recovery:OFF
de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
>>
```

To put a new set of parameters into the RAM memory in CU type **I Enter**:

```
Init values
Are you sure? (Y/N)
>>
```

To affirm this, press **Enter** (the same as **Y Enter**)

```
put   UNIT 0 O.K.
>>
```

The `put` operation was successfully finished, press **Enter** for displaying the menu...

To move faster through the menu it is possible to write string commands, for example **Ue Enter Enter** written from the main menu jumps directly to the menu `(U)nit (e)dit`

1.2. CU Configuration Services

Configurable parameters of the CU are divided into independent groups called configuration structures. A separate submenu exists for each structure type (there can be more configuration structures of the same type, e.g. five structures of Node parameters in the MR25). Within an CU, each configuration structure can be found in three different representations:

- working parameters stored in RAM

  These parameters are the working ones, so any change immediately influences the CU operations. When the CU power supply is switched off, or the CU software is restarted due to any reason, the RAM parameters are lost.

- initial parameters stored in FLASH memory

  The FLASH memory keeps its content even without a power supply, so these parameter sets are used for long-term storage of the CU configuration. The FLASH configuration structures are copied to RAM structures during an CU software restart.

- default (backup) parameters stored within the software code

  These are read-only parameter sets, used as a basis for the CU setup, or as a "last resort", when both RAM and FLASH parameters are destroyed.

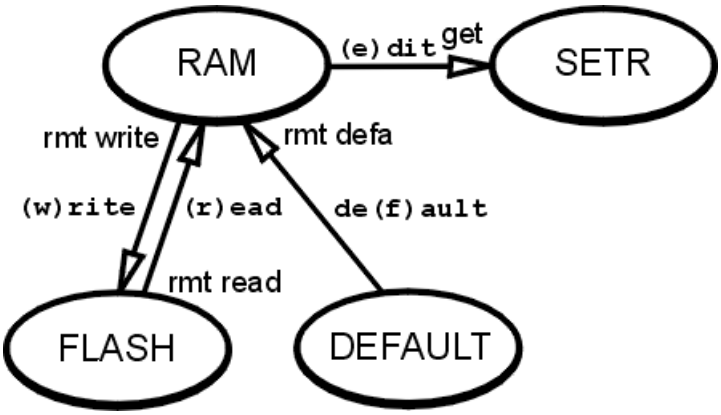

*Fig. 1.2: Before making command **edit** (i.e. in menu, which contains command edit*

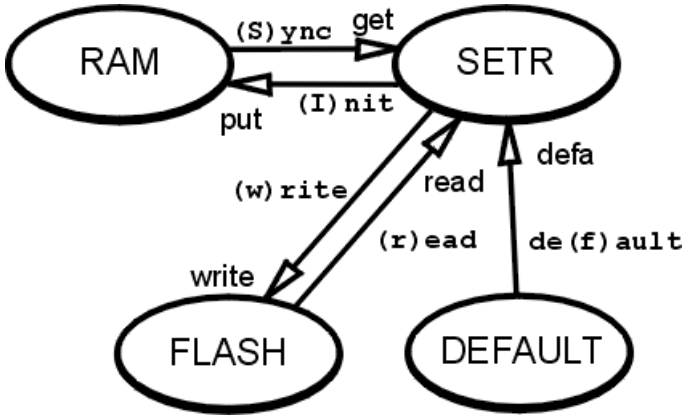

*Fig. 1.3: After making command **edit** (menu doesn't contain command edit)*

Program Setr.exe configuration menus provide two basic levels, from where configuration services can be executed. Commands **(r)ead, (w)rite, de(f)ault, (I)nit, (S)ync** and **(e)dit** , some of them existing in both menu levels, call these services. A configuration service deals only with the structure (or the group of structures) appropriate to the submenu from which it has been invoked.

In the said pictures, RAM, FLASH and DEFAULT represent the respective configuration structures within the CU and SETR stands for the temporary buffer in Setr.exe application. The content of this buffer can be displayed on the Setr.exe screen, and individual parameters can be edited in respective parameter submenus.

To see which menu command calls which service and which configuration structure it deals with, please check the said pictures. The words `(w)rite, (r)ead, de(f)ault, (S)ync, (I)nit, (e)dit` represents the commands executed in the Setr.exe.

The above mentioned rules are valid for the individual configuration structures. When we need to put the complete CU into it's original state, we can use the submenu Configuration, see the chapture Chapter 14, *Menu (c)nf*.

1.3. Useful tips

When issuing a configuration command (i.e. **(I)nit, (S)ync, (e)dit, (r)ead, (w)rite** or **de(f)ault)**, always wait for the `get (put, write...) service O.K.` response. If nothing comes in due time, repeat

the command. As the edit command moves you to the next menu level even without receiving the `get service O.K.` response, you can use **(S)ync** to repeat this command.

While working with a remote CU, do not use the **ESC (q) Enter Enter** sequence instead of the **Init** command. If the `put service O.K.` response fails to appear, you need to stay in the same menu level to be able to repeat the command with the same content of the temporary buffer.

*Always think twice before issuing the **(I)nit** or **(w)rite** command to a remote CU ...*

2. Hardware

The Hardware menu is appointed for reading only. It contains the basic production data concerning the CU. From Main menu type `He Enter Enter`

```
HW data:
product (t)ype:MR400 su(b)type note:
HW (v)ersion:3 s(u)bver:0
(s)erial:4837383 (h)ex:0049D007
prod date (d):11 (m):3 (y) 1970+:34
system (c)lock:66666667
de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
>>
```

Meaning of the menu items:

product (t)ype:	marking of the product type, e.g. `MR400`
su(b)type note:	possible additional information
HW (v)ersion:	main version number of the unit hardware
s(u)bver:	sub–version number of the CU hardware
(s)erial: 4837383	production serial number in the decimal form should be equal to the serial typed on the production label
(h)ex: 0049D007	production serial number in hex, it is also the MORSE address of the node 0
prod date	date of manufacture in the form day, month, year
system (c)lock	approximate frequency of main system clock (Hz), the exact value see the menu `(s)ervice`
de(f)ault	This and next commands are not used it the Hardware menu. The valid values appear after restart or by using the command `(r)ead`
(q)uit	return one step back

3. Communication Unit

The Communication unit menu contains the user parameters concerning the all CU. Enter from Main menu using the command `Ue Enter Enter`.

```
Communication unit:
service (d)est:00000000 (N):0
(U)cc limit:130 * 0.1V  (h)yst.:2 * 0.1V
(s)tatus:0000 (P)SWD:OFF (L)OGW:OFF (M)INP:OFF
(l)og period:86400
Time (Z)one for DST:EU Time zone DI(F)F:3600sec
           obsolete(<605) Time (z)one:69136sec
Summer time (all to zero - off):
start (1)month:3 (2)day:27
end   (3)month:10 (4)day:30
Time sync:
 (i)n:0 Sr(c):00000000 (p)eriod:0
(O)ut:0 ds(t):00000000 p(e)riod:0
P(A) dir:0000 dat(a):0000
P(B) dir:0000 dat(b):0000
Power saving:
alar(m) time:0s m(o)de:0000
(R)adio pll recovery:OFF
de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
>>
```

service (d)est: 00000000 — The destination address for system reports. Obsolete, replaced by the menu `ise` setting.

(N):0 — Originating node for system reports. Obsolete, replaced by the menu `ise` setting.

(U)cc limit: 130 — for MR400 until fw685 and for MR25 only - intermediate voltage for `Ucc (h)yst`, here 13.0V.

Ucc (h)yst.: 2 — for MR400 until fw685 and for MR25 only - hysteresis added and subtracted to the `(U)cc limit` gives the up and down supply voltage for the writing to the error log, here from 12.8 to 13.2V

(s)tatus: 0000 — four hex. characters are displayed, individual bits are set by independent menu items as follows:

(P)SWD: OFF — (s)tatus = 0x0100

- `OFF` — for service purposes - set OFF

(L)OGW: OFF — (s)tatus = 0x0002

- `ON` – All statistic logs are closed and written to the flash memory while the unit supply voltage drops out.
- `OFF` – Only the system error 1999 is recorded during the supply voltage drop-out.

(M)INP: ON — (s)tatus = 0x0004, sends minute pulse on SCC2-RxC pin, i.e. Cannon DSUB37F, pin 35

(l)og period: 86400 — The time period (in seconds) after which the statistic logs are closed and recorded in the flash memory, and are then available in the log history (3 periods in all).

Summer time

From SW 5.34 version — automatically setting the interval of summer time validity. The line `start` contains the month and day, when summer time starts, the line `end` contains the month and day when it ends In this range the difference between **GMT** maintained in the CU and the local time, saved in the `Time (z)one`, is increased by 1 hour. The value of **GMT** remains without changes and the local time visible in the menu `(s)ervice (t)ime` or in the menu `(s)ervice (s)tatus` is shifted.

The time changes in such a way that in the day designated by line `start` at 02:00:00 the local time is shifted to 03:00:00 and in the day designated by line `end` at 03:00:00 the local time is shifted back to 02:00:00. Considering the inconstancy of the start and end data for the validity of summer time approximate data is inserted into the items start and end which can be corrected individually.

When the year entry is changed in the CU when setting the time using the `(s)ervice (t)ime` menu, then new initialisation should be performed in the menu `(U)nit (e)dit`.

Time (Z)one for DST: EU — Time Zone Calendar for Day Light Savings

- (E) EU — automatically setting, use `(F)` parameter only, the dates `(1),(2),(3),(4)` are counted automatically for the respective year
- (N) NONE — manual Light Saving setting, use `(z)` parameter and `(1),(2),(3),(4)` params

Time zone DI(F)F: 3600 — difference between the local time and GMT, without influence of Daylight Savings

Time (z)one: -3600 — The number of seconds which should be added to the local time to get GMT.

Time sync

Time synchronisation messages schedule. Time-sync messages are used to synchronise the calendar time in adjacent units in a MORSE network. Two modes are used:

1. A unit periodically requests the calendar time from a node in the neighbouring unit. This mode is mostly used, as it allows more units to be synchronised from one source.

   (i)n:0 — No of the node which originates the time-sync request message

   Sr(c):0000000 — address of the time source

   (p)eriod:0 — period of time-sync requests (in seconds)

2. A unit periodically transmits the time-sync message to the configured destination, to synchronise the calendar time in the unit which contains the destination node address. This mode saves the request message, but it is limited to one destination synchronised from one source.

(O)ut:0 — No of the node which originates the time-sync message

ds(t):0000000 — an address in the unit to be synchronised

p(e)riod:0 — period of time-sync messages (in seconds)

Parameters for development purposes

P(A) dir:0000 — direction switching (1 means output)

dat(a):0000 — data (output pins are set, input pins are read) čteny)

P(B) dir:0000 — direction switching (1 means output)

dat(b):0000 — data (output pins are set, input pins are read) čteny)

(R)adio pll recovery:OFF - the service parameter

4. Radio part

The Radio part menu only applies to communication units (CU) where the radio hardware is an integral part of the unit. The most typical example of such a unit is the MR400 or MR25.

Sub-menus Radio Hardware and Radio Parameters deal with the configuration stored primarily within the radio part of the hardware. Still both of these configuration data structures are kept in the communication unit RAM and SRAM or FLASH memories the same way as the ``normal'' ones.

The relations between memory modules in Radio menu, compare fig. of memory in *MORSE main menu*.

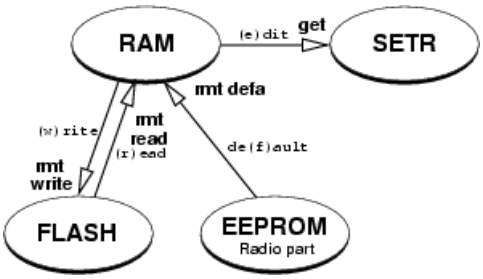

Fig. 4.1: Before edit execution (i.e. in the menu, which contains the command edit).

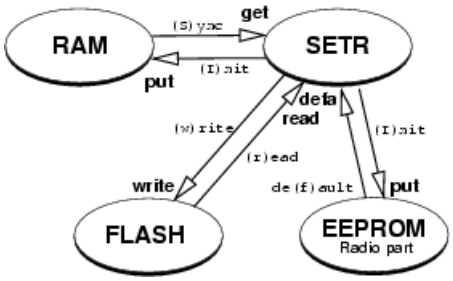

Fig. 4.2: After the edit function.

Nevertheless it is important to be aware of the fact, that the actual working parameters are the ones stored in the radio part of the hardware. Due to this fact, some commands in the above mentioned menus behave differently from "normal" configuration menus. Utmost care must be taken when changing Radio part parameters in a remote unit, as the risk of irreparable communication breakdown is considerably high.

The specific commands behaviour is discussed in the appropriate paragraphs.

4.1. Radio hardware

Similar to the unit HW parameters menu, this menu can be normally used only for displaying the HW configuration of the radio part. The actual working set of Radio HW parameters is stored in the radio part memory and cannot be changed from this menu. These original parameters are read by both DEFAULT commands (before and after the `(e)dit` - see section Section 1.2, “CU Configuration Services”. These commands can be performed at anytime, and must always be applied before any change in Radio part has taken place.

Caution

The copy of the radio part HW parameters in unit RAM is used e.g. as a basis for calculations of by SETR displayed frequencies. Thus a change of Radio part HW parameters can result in an incorrect display of these frequencies and possible the wrong setting and saving of it.

```
MORSE main menu:

(H)W   (U)nit (R)adio
(N)odes  s(D)r   (B)c
(S)CC   R(F)C   (E)th
(A)rt   r(T)ab  (c)nf
(m)isc  NA(G)

(s)ervice      d(i)ag
(p)ath        (?)help

(o)ld cnf menu

(q)uit
```

type **R Enter**

```
Radio part:
(h)w data
(p)arameters
(c)alibration
(q)uit
>>
```

type **he Enter Enter**

```
Radio hardware:
product (t)ype:MR25R2
(s)erial:8543
(H)W ver:40  subver:1
flags:0000
S(W) ver:19
prod dat(e) d:28 (m):2 (y) 1970+:35
check date D:0 M:0 Y 1970+:30
MR(2)5R  MR(9)00
de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
>>
```

This menu is meant for reading only.

(t)yp product (t)ype: MR25R — type of the radio part

```
  product type
(G) GPRS      - MG100 = modem with GPRS module instead of radio
(2) MR25R2    - MR400, MR300, MR160
(M) MR25R     - MR25
(9) MR900     - MR900
(N) NO RADIO  - MC100, MCM302, MORCE, WALRUS
```

(s)er (s)erial:8543 — Radio part serial should be equal to the serial typed on the production label

(H)W (H)W ver:40 — version number of HW

flag flags:0000 — the flags inform about some features of the radio part, e.g.:

flags:0000	standard version
flags:8000; narrow band	channel bandwidth 12,5 kHz (see `FMe` menu)
flags:0010; custom ch.spacing	fifth bit from the right having the value 1 indicates that the TX and RX frequencies can be chosen with special space: — 6,25 kHz at parameter `Rhe 2 (c)h spacing:125*100Hz` using parameter `Rpe Frequency off(s)et` — 10 kHz at parameter `Rhe 2 (c)h spacing:100*100Hz`

S(W) S(W):19 — version number of SW

dat(e) prod dat(e) — date of manufacture – day, month, year

check check date — date of output check – day, month, year

MR(2) MR(2)5R — frequency setting for MR25 and MR400

```
MR25R control:
(c)h spacing:125*100Hz  switching r(a)nge:32000*100Hz
IF        - (T)X:565000*100Hz  (R)X:-450000*100Hz  RX (2)nd:4550*100Hz
base freq - T(X):4250000*100Hz  RX(b):4250000*100Hz
(q)uit
>>
```

(c)h (c)h spacing: 125 — the values 125 or 100 represent the frequency step of RX and TX main synthesizers 12,5 kHz or 10 kHz. The working frequency in Rpe menu can be written with this step, see also the item flags. This parameter does not represent the channel bandwidth, see `FMe` menu, parameter FMe 9s)

r(a)n switching r(a)nge: 32000 — the range from the base frequency (in 100 Hz), where Radio part is adjusted to operate and RF parameters are guaranteed. Receiving and transmitting ranges can be adjusted separately, i.e. split operation is possible

IF-(T) IF-(T)X: 565000 — transmitter intermediate frequency (in 100 Hz)

IF-(R) IF-(R)X: -450000 — receiver intermediate frequency (in 100 Hz)

(2)nd IF-RX (2)nd: 4550 — receiver second intermediate frequency (in 100 Hz)

T(X) base freq T(X): 4250000 — transmitter base frequency (in 100 Hz)

RX(b) base freq RX(b): 4250000 — receiver base frequency (in 100 Hz)

MR(9) MR(9)00 — frequency setting for MR900

```
MR900 control:
base freq (0):8695250*100Hz
base freq (1):8695250*100Hz
(q)uit
>>
```

freq (0) base freq (0): 8695250*100Hz — working frequency (in 100 Hz)

freq (1) base freq (1): 8695250*100Hz — second working frequency, if used

4.2. Radio parameters

Some of the Radio part parameters stored in its memory have to be accessible for changing by the main unit software (namely RX and TX frequencies, output power and status bits). These parameters are concentrated in the Radio parameters configuration structure and can be edited in the following menus.

The same way as in the Radio HW parameters menu, both DEFAULT commands read all data from the Radio part memory. The most important difference from the Radio HW menu is that anytime the data in the unit RAM are changed (e.g. after `(I)nit` command), new values of power level and synthesizer rates are calculated and written to the Radio part memory. These calculations use some values from the Radio HW parameters structure, so it is absolutely necessary to have correct values in the Radio HW structure in the main unit RAM before making any change to Radio parameters. If in any doubt it is strongly recommended to enforce the `de(f)ault` command from the `Rh` menu (see `(R)adio (h)w` data for more details).

From MORSE main menu type **Rpe Enter Enter**:

```
Radio parameters:
(T)X:4264750*100Hz  (R)X:4264750*100Hz
Frequency off(s)et:NONE  :0
Power - (l)evel:7  mW:275
(c)heck period:0
(1)-tx  (2)-rx
de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
>>
```

(T) (T)X:4264750 - transmitter operating frequency (in 100 Hz). Only these frequencies can be saved, which correspond to the frequency grid according to the parameter (c)h

spacing (Rhe 2c menu). Check the right saving using (r)ead, (S)ync. Since 12/2008 is the frequency step indicated also in the production code[1].

(R) (R)X:4264750 - receiver operating frequency (in 100 Hz)

(s) Frequency off(s)et:NONE :0 - active at radiomodems having the frequeny step 6.25 kHz only, see (c)h spacing and production code[2]. It increases the frequencies set by the parameters (T)X, (R)X by 6.25 kHz.

(l) Power - (l)evel:7 mW:275 - transmitter power level is set by hex number from 0 to F. The displayed value in mW is calculated during the `(S)ync` command execution. To see the correct value following any change, `(I)nit` and then `(S)ync` commands have to be performed. Use `(w)rite` for permanent saving.

(c) (c)heck period:0 - for service purposes

(1),(2) (1)-tx (2)-rx - alternative frequencies used by the mobile mode. More informations in the manual MORSE Guide 2[3]

```
Allowed TX freqs: (*100Hz)
(0) :4264750  (8) :0
(1) :4265000  (9) :0
(2) :0  (A) :0
(3) :0  (B) :0
(4) :4265750  (C) :0
(5) :0  (D) :0
(6) :0  (E) :0
(7) :0  (F) :0
(q)uit
>>
```

The items (0),(1)...(F) represents the individual frequency channels. The parameters are filled in two ways:

- Bases of mobile network - CU runs on the frequency written by the parameters (T)X, (R)X. The nonzero value of choosen parameters (0),(1)... label the frequency channels of Bases situated on common location.
- Mobile - parameters (T)X, (R)X doesn't need to be filled in. The frequencies written in choosen items (0),(1)... define the frequency channels where the Mobile can be retuned.

4.3. Radio calibration

The Radio part provides the main unit with "raw" values from the A/D converters, which measure some important analogue signals (Received Signal Strength, internal temperature, supply and PLL voltages). To get the actual values in dBm, Volts, degrees etc., a set of computing constants is needed. A similar set of constants is needed to get the transmitted power value in mW for the configured power level number.

[1] http://www.racom.eu/eng/download/morsecode.html
[2] http://www.racom.eu/eng/download/morsecode.html
[3] https://www.racom.eu/eng/support/morse-m2/index.html

Caution

All these constants are hardware related and are written to the unit flash memory during the production process. The Radio calibration menu allows for displaying and editing of all these constants. The default values provide a set of initial values to start the calibration process during production or maintenance.

In normal situations, it is strongly recommended not to change anything in this menu. Especially any use of (w)rite commands can result in the need to send the affected unit to the manufacturers premises for re-calibration.

From `MORSE main menu` type **Rce Enter Enter**

```
Radio calibration:
RSS  - (a)c0:34496  (b)c1:97
t(e)mp - c0:0  (g)c1:300  (h)c2:0
AF v(o)lt c0:22
main pll - (T)X c0:111  (R)X c0:111
loc  pll - (t)x c0:111  r(x) c0:111
(U)cc volt c0:111  RSS (v)olt c0:20
power level constants
(0):0 (1):0 (2):0 (3):0 (4):0 (5):0 (6):0 (7):20000
(8):0 (9):0 (A):0 (B):25700 (C):0 (D):28399 (E):31700 (F):37000
de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
>>
```

In the given example of `power level constants` it is visible, that this modem (`(H)W ver:30`) is ready for using Power levels `7,B,D,E,F`. The modems `(H)W version:40` can use all Power levels from `0` to `F`.

5. Nodes

The node is the basic organisational unit of the MORSE network. It is defined by entering the address and works independently to others nodes. In one CU it is possible to define up to 4 nodes + 1 node for service purposes.

Basic routing parameters of the CU are set in this submenu - node addresses, channels assigned to nodes outputs, routing tables assigned to individual nodes etc.

From Main menu type `Ne Enter Enter`

```
  Nodes:
                                retab
Nid|address |M | u   s | L   N |l w n g H|sTO Err  Cent vTO hTO
(0) 0049D007     -  S00| -  R00|0 0 0 0 -| 15 SERV  OFF 304  30
(1) 690F0003    S00 S00| -  R01|0 0 0 0 -| 15 SERV  OFF 304  30
(2) 690F0013    S01 S00| -  R02|0 0 0 0 -| 15 SERV  OFF 304  30
(3) 690F0023    S02 S00| -  R03|0 0 0 0 -| 15 SERV  OFF 304  30
(4) 00000000    S03 S00| -  R04|0 0 0 0 -| 15 SERV  OFF 304  30

de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
```

type `1 Enter`

```
Nodes:
(a)ddr:690F0003 (M)ultiaddressing
(u)c:S00 (s)c:S00 R(L)ink: -  R(N)et:R01
(l)o:0 (w)i:0 (n)e:0 (g)l:0
(H)rt:OFF
Store (T)O:15
(C)enter:OFF e(x)tended:OFF (v)alidity TO:19*16 s (h)istory TO:30
(E)rr:SERV
(I)nit (W)rite
(q)uit
>>
```

(a)ddr: 690F0003	the node address, (node 0 address is equal to the unit serial and cannot be changed)
(M)ultiaddressing: L on/off	when ON, packets routed to Link output go to the user output of the node
(M)ultiaddressing: N on/off	when ON, packets routed to Net output go to the user output of the node
(u)c:S00	channel SCC0 is connected to the user output from the node, for no user output type `un Enter`
(s)c:S00	channel SCC0 is connected to the service output to which the service cable is plugged. The MR400 uses SCC0 as the service channel, MR25 uses SCC2.

R(L)ink:- no channel is connected to the retranslation link output

R(N)et:R01 channel RFC1 is connected to the retranslation net output

(I)o:1 assigning of routing tables to the node, local table, “0” means no table used

(w)i:0 wide table

(n)e:0 net table

(g)l:0 global table

(H)rt:OFF table for “hierarchy” routing, the state OFF is “-”, see the chapter Chapter 6, *Morse Redundancy Solution*

Store (T)O:15 The time (sec) within which a packet has to leave the unit. If exceeded, the packet is discarded and an error message is generated.

(C)enter:OFF When ON, the node is switched to Mobile Centre mode. In this mode the node can keep information about the mobile units location, i.e. for each active mobile address the currently valid base address is kept. For more information see www.racom.eu, Support, Mobile Network[1].

Important

The number of Mobiles registered is limited according to type CU used as Center:

- MR25, MR25ET, MCM302 - 64 Mobiles
- MR400, MR300, MR160, MC100, MG100, MR900 - 450 Mobiles
- SW Walrus - 2000 Mobiles

e(x)tended: OFF in development

(v)alidity timeout: 19*16 s When the Mobile Centre mode is on, this parameter sets the time (in seconds) for which the base address information assigned to a mobile address is regarded as valid (here 19×16=304 s).

(h)istory TO: 30 timeout (sec), over which the (C)enter checks and blocks packet duplicity from the mobile station; the packet must contain the net number, 0 = the check is off

(E)rr:NONE NONE/MORSE/C92/DEBUG/SERV/S+U

The modes of error messages:

The Node can generate the error message for two reasons: the communication error (the ACK from the opposite CU was not received) or the configuration error (e.g. the attempt to send the packet in non-exist channel). The error messages can be generated for all packets or for some packets only.

[1] https://www.racom.eu/eng/support/mobile_station.html

	Packet type	*Error type (communication,configuration)*
MORSE	all	communication error
DEBUG	all	all
SERV	service request	all
S+U	service request user data, prot data	all
C92	obsolete, for the compatibility with RD300FS (packet type C5)	communication error

6. Morse Redundancy Solution

This part of the menu is under development. The original meaning of the abbreviation SDR (Semi Dynamical Routing), now known as Morse Redundancy Solution, expresses the function of the automatic selection of the communication route according to the situation in the network. It contains three independent parts:

```
Morse Redundancy Solution:
(N)ode extensions
(G)lobals
(H)ierarchy Routing Table
(q)uit
>>
```

6.1. Node extensions

This menu contains the list of variant routing tables. They are used in the Morse Redundancy Solution system also at choice of spare path according to the menu `DGe oN`.

With choice DNe we get the table, where the variant routing table numbers can be written similarly like in the menu Ne:

```
  Node Extensions:
     retab
Nid |l w n g| Mode
(0) |0 0 0 0| 0
(1) |0 0 0 0| 0
(2) |0 0 0 0| 0
(3) |0 0 0 0| 0
(4) |0 0 0 0| 0
```

Go in the submenu by writing e.g. 1. Here switch ON the parameter (K)eep lines. It is necessary in the mode NEXT NODE for the test packet sending.

```
Node Extensions:
(l)o:0 (w)i:0 (n)e:0 (g)l:0 (H)rt:0
(K)eep lines:OFF
```

The item (H)rt:0 is under development.

6.2. Globals - Mobile mode

Globals menu contains the parameters for the Mobile mode and for the Next Hop mode which can switch the backup communication line. Since version 9.0.0.0 the parameters for old mobile mode are shifted here from the menu FPe. The manual MORSE Guide 2 [1]describes this modes.

From main menu choose `DGe Enter`:

```
>>DGe

Dynamic routing: Globals
m(o)de:MM ROUTER           ... selection Router/Base/Mobile
validity (t)imeout:120sec  ... validity of entry in dynamic table
(N)id:1                    ... selection of node for this mode
Parameters:
mm (m)obile                ... other parameters for Mobile station
mm mob(i)le obsol          ... older Mobile mode, in menu FPe formerly
mm ro(u)ter                ... other parameters for Router
mm (b)ase                  ... other parameters for Base

ob(s)olete
de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
>>
```

The parameter `m(o)de` offers the mode:

```
>>DGe o

  mode
(0) OFF             ...standard mode, stationary routing
(M) MM MOBILE       ...mobile mode, mobile CU
(O) MM MOBILE OBSOL ...old mobile mode, mobile CU
(L) MM BALANCED     ...in development
(R) MM ROUTER       ...mobile mode, router
(B) MM BASE         ...mobile mode, base
(N) NEXT HOP        ...backup path, Next Hop mode
(K) NEXT HOP+KEEPS  ...backup path, Next Hop+Keep mode
```

The parameters for both mobile mode follows.

Router

```
>>DGe u

Router mode parameters:
Mobile:  (b)ase:690F8000  (m)ask:FFFF8000  (c)entre:690F7E05
Load treshold: Lo(w)/M-L:200promile  Hi(g)h/H-M:600promile
Load meas (T)au:5000ms  (o)wn load treshold:500bps
debug via (S)ystem channel:OFF
```

[1] https://www.racom.eu/eng/support/morse-m2/index.html

```
(q)uit
>>
```

(b) (b)ase:690F8000 - reference address for definition of Mobile stations

(m) (m)ask:FFFF8000 - mask for definition of Mobile stations

(c) (c)entre:690F7E05 - address of central application

(w) Lo(w)/M-L:200promile - evaluation of RF load, less than 20% is level 3

(g) Hi(g)h/H-M:600promile - 20% to 60% is level 2, more than 60% is level 1

(T) (T)au:5000ms - auxiliary parameter for evaluating RF load

(o) (o)wn load threshold:500bps - auxiliary parameter for evaluating data load

(S) (S)ystem channel:OFF - switches on transmission of debug messages to the System channel `ise 1`

Base

```
>>DGe b

Base mode parameters:
Mobile:  (b)ase:690F8000  (m)ask:FFFF8000  (c)entre:690F7E05
Base info (t)imeout:30sec  base-centre (e)cho timeout:0sec
Load treshold: Lo(w)/M-L:200promile  Hi(g)h/H-M:600promile
Load meas (T)au:5000ms  (o)wn load treshold:500bps
debug via (S)ystem channel:OFF
(q)uit
>>
```

In addition bases use the following parameters:

(t) Base info (t)imeout:30sec - interval for sending packet `base info` to the RF channel

(e) base-centre (e)cho timeout:0sec - interval for sending test pings to the centre - if no response returns transmission on the RFC is interrupted and the Mobile station switches to another Base

Mobile station

```
>>DGe m

Mobile mode parameters:
Apps:  (b)ase:690F7E00  (m)ask:FFFFFE00  (c)entre:690F7E05
Base:  (B)ase:690F0000  (M)mask:FFFF8000
(P)assive:OFF  (h)ome frequency id:1
(r)etune timeout:30sec  (k)eep timeout:30sec
(p)ifka timeout:60sec
Load treshold: Lo(w)/M-L:200promile  Hi(g)h/H-M:600promile
Load meas (T)au:5000ms  (o)wn load treshold:500bps
debug via Event (L)og:OFF
debug via (S)ystem channel:OFF
```

```
Retune method parame(t)ers
(q)uit
>>
```

(b) Apps: (b)ase:690F7E00 - reference address for definition of applications

(m) Apps: (m)ask:FFFFFE00 - mask for definition of applications - packets sent to these addresses renew records in dynamic tables of Bases and Routers and prevent the transmission of a keep packet for the period DGemk

(c) (c)entre:690F7E05 - address of central application - keep packets leave to this address

(B) Base: (B)ase:690F0000 - reference address for definition of Bases

(M) Base: (M)mask:FFFF8000 - mask for definition of Bases - only these addresses are accepted as a Base

(P) (P)assive:OFF - with this setting station MR switches the antenna through the duplexer to MR or MW during a change of operation between the narrow and wide band, the signal is on SCC3

(P)assive: ON - MR does not give a signal for the duplexer

(h) (h)ome frequency id:1 - number of the channel on which the Mobile station begins to search for a Base upon start up or upon a loss of connection

(r) (r)etune timeout:30sec - the Mobile station does not switch between various frequencies before this time has elapsed; when switching between Bases from various locations on the same frequency this limitation does not apply (so-called rapid switch)

(k) (k)eep timeout:30sec - in this interval the Mobile station sends a maintenance packet to the Centre to address DGemc

(p) (p)ifka timeout:60sec - the parameter Crit is reduced for this time period for the Base with which problems occur in communication - this results in switching to another Base, if available

(w) Lo(w)/M-L:200promile - limit for distinguishing the load levels 3 and 2 on the RF channel - for a load on the RFC for less than 20% of the time the level is 3

(g) Hi(g)h/H-M:600promile - limit between level 2 and 1 - a load greater than 60% gives level 1

(T) Load meas (T)au:5000ms - auxiliary parameter for evaluating RFC load

(o) (o)wn load threshold:500bps - auxiliary parameter for evaluating data transfer load

(L) debug via Event (L)og:OFF - record to debug messages to Event log

```
(0) OFF          ... off
(1) PROBLEMS     ... erroneous configuration of frequencies,
                     loss of location
(2) +RETUNE INFO ... (1)+ retuning to another Base
(3) FULL DIAG    ... all messages
```

(S) debug via (S)ystem channel:OFF - messages sent to System channel 1 (menu ise)

```
(0) OFF   ...off
(1) INFO  ...message about switching Bases, etc
(2) TRACE ...message sent with an interval of 1 sec
(3) FULL  ...table mrm sent with in interval of 1 sec,
             heavy channel loading
```

(t) Retune method parame(t)ers - parameters which are important for retuning to another Base:

```
>>DGE mt

Retune method parameters:
Retune m(o)de:RSS/DQ
(D)q weigh [mode RSS/DQ]:700promile
(M)edia type setting
Criterium (h)ysteresis:20promile
Dq  transform curve: Kx(1):15  Ky(2):950promile
Rss transform curve: Kx(3):85dBm  Ky(4):800promile
Problem meas (T)au:15000ms  (P)roblem treshold:900promile
Problem (u)ncertainity treshold:100 [1/100]
Bobeks:
(r)ss:150promile  (p)ifka:250promile
(c)onnection:250promile
pac(k)ets:650promile
(q)uit
>>
```

(o) Retune m(o)de:RSS/DQ - method of processing RSS and DQ when evaluating a Base

```
(0) RSS/DQ ...DQ has a weight according to DGemtD,
              the weight of RSS is an increment to 1000
(1) DQ     ...other weights, under development
```

(D) (D)q weigh [mode RSS/DQ]:700promile - weight DQ, weight RSS is 1000 - D

For all Bases at one location there is taken a common value of RSS and a common value of DQ.

(M) (M)edia type setting - transmission media in individual channels are characterised by a four-bite character, for example:

```
MR:  0010 = 0x02
MW:  0101 = 0x05

meaning of bits from the left:
bit  3   - res
bit  2   - 0 channel preferred for transfer of individual packets
              (packet)
           1 channel preferred for transfer of a large amount
              of data (connection)
bity 1,0 - 00 ETH
           01 wide - wideband
           10 narrow - 25 kHz band
           11 microwave channel
```

(h) Criterium (h)ysteresis:20promile - parameter Crit must change by at least this value in order to induce a change in channel

(1) Dq transform curve: Kx(1):15 - for converting DQ values into a value used in the calculation a pair of abscissae defined by coordinates DQ/value are used: 0/0, Kx(1)/Ky(2), 31/1000.

(2) Dq transform curve: Ky(2):950promile - second coordinate of conversion abscissae for DQ

(3) Rss transform curve: Kx(3):85dBm - definition of conversion abscissae for RSS

(4) Rss transform curve: Ky(4):800promile - second coordinate of conversion abscissae for RSS

(T) Problem meas (T)au:15000ms - auxiliary parameter for the calculation of parameter Problem, a longer time increases the influence of historic values of the parameter Problem and slows down the reaction to new events

(P) (P)roblem treshold:900promile - parameter Problem is calculated from more indicators, it rises particularly during unsuccessful communication with the centre. If the limit `(P)roblem treshold` is exceeded then Pifka status is set for this channel. This results in the Mobile station attempting to use a more suitable Base.

(u) Problem (u)ncertainty treshold:100 [1/100] - together with parameter Problem the parameter Problem uncertainty is also calculated. Uncertainty drops with fresh data about successful or unsuccessful communication. Uncertainty cannot be greater than the limit set here in order to allow for the application of Pifka.

(r) (r)ss:150promile - all criteria at the location are multiplied by this value (=0.15) with the exception of channel home (DGemh) in the case where RSS weakens to below -95dBm. This results in switching to channel home on which setting up communication to the next location is easier.

(p) (p)ifka:250promile - the criterion of the channel to which status pifka was applied is multiplied by this value (=0.25). This results in the Mobile station switching to another Base if available.

(c) (c)onnection:250promile - during the transfer of "connection" type data, i.e. a large volume of data, the criterion on channels unsuitable for this transfer is multiplied by this value (=0.25). An example of this are channels with parameter DGemtM0 = 2, i.e MR160.

(k) pac(k)ets:650promile - during the transfer of "packet" type data, i.e. short packets, the criterion on channels unsuitable for this transfer is multiplied by this value (=0.65). An example of this are channels with parameter DGemtM0 = 5, i.e. MW160.

The result of the last two parameters is that narrowband (MR) modems deal with the transfer of short data and wideband modems (MW) remain available for the transfer of large volumes of data. These transfers to various Mobile stations can occur simultaneously.

Mobile obsolete

The old mobile mode parameters are shifted here from the menu FPe. Detailed description in the manual MORSE Guide 2[2]

```
>>DGe i
```

[2] https://www.racom.eu/eng/support/morse-m2/index.html

```
Mobile obsolete mode parameters:
(b)ase:690F0100  (m)ask:FFFFFF00  (c)entre:690F0233
(k)eep timeout:61sec
(q)uit
>>
```

(b) (b)ase:690F0100 - comparative address for choice of base stations

(m) (m)ask:FFFFFF00 - if any station which can be received on the RF 0 channel is to be taken as base, masked parts of the (b)ase parameter and the base station address must be equal, e.g. 690F0105

(c) (c)entre:690F0233 - address of central station, where this mobile has to report to

(k) (k)eep timeout:61sec - period (sec) for reporting to the central station (0 means no periodical reports are sent)

6.3. Hierarchy routing table

The routing based on hierarchy routing tables is a variety to the standard routing tables. It is implemented since version 7.60. It is activated from the Nodes menu choosing e.g. `Ne 1H1`.

The `(H)ierarchy Routing Table` menu is in a development stage and it's control is unnormalized. It works with the S-RAM memory only analogous to the common routing tables. The commands `de(f)ault (r)ead (w)rite` in the DHe menu are used only for the reading and writing.

```
  Hierarchy routing table:
Hid |-
(0)| 5
(1)| 5
(2)| 5
(3)| 5

de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
>>
```

After selecting `He Enter` it is necessary to read the contents using command `r Enter`. Now select one of the tables (0) to (3) `by command 0 Enter`:

```
Hierarchy Routing Table
    (b)ase   (m)ask   (t)oa    tm(a)sk
(0) 690F0000 FFFFFF00 00002100 FFFF00FF
(1) 00000000 FFFFFFFF 00000000 00000000
(2) 00000000 FFFFFFFF 00000000 00000000
(3) 00000000 FFFFFFFF 00000000 00000000
(4) 00000000 FFFFFFFF 00000000 00000000
```

Use a combined command to edit, e.g.`0b690F0000 Enter`. For saving we have to return by one level `q Enter` and save using command `w Enter`.

If the HRT mode is switched on in the Nodes menu then when routing HRT is browsed first and only then the routing tables. HRT 0 to 3 assigned to the node contains max. 5 items in rows (0) to (4), which are browsed from top to bottom.

Item `(b)ase` must be entered in such a way that it has a 0 on those bits on which 0 has the respective mask `(m)ask`.
Item `(t)oa` must have a 0 on those bits on which the mask `tm(a)sk` has the value 1 (or F in the hexadecimal meaning).

The destination address is compared with item `(b)ase` in the scope of `(m)ask`. If destination agrees with the base then a new address Next Node is created for routing in such a way that a part according to the mask `tm(a)sk` is taken from destination address and that part where `tm(a)sk` has zeros (after bits) is added from `(t)oa`. Then HRT is exited and the operation with routing tables is skipped and routing is completed with the Next Node address obtained in this way.

If destination and `(b)ase` do not agree the next row of HRT is tested if it is filled in. After exhausting all the possibilities offered by the HRT table the routing solution continues by using routing tables.

7. Broadcasting

Any packet (message) entering the MORSE network can be labelled as a broadcast one. Either the user labels selected packets, or it is possible to set labelling of all packets coming through the user part of the respective CNI (Channel to Node Interface). Once a broadcast packet enters the network, it is spread around following `(B)c` menu settings of all the nodes that receive it.

With careful settings in all network nodes it is possible to achieve the ultimate goal of broadcasting - to deliver a message simultaneously to all network members – even in large networks with a complex architecture.

When a broadcast packet is transmitted in the RF channel, every receiving CU delivers it to the respective node (to all 5 nodes in radio part). There it is checked for the source address first. If the source address corresponds with the node settings (src base & mask), the packet is multiplied and delivered simultaneously to different node outputs according to the `(B)c` menu settings.

For example, let us have the user and net node outputs switched on. The received (and checked) broadcast packet is then delivered to the user channel and simultaneously to the retranslation network channel. The latter is typically the RF channel, so in this example the packet will be transmitted on the RF channel again. Using the configuration described above, even broadcast packets can exploit store-and-forward repeaters. Of course an effective algorithm for discarding repetitive packets is implemented.

From MORSE Main menu type **B e Enter Enter** .

```
  Multicast routing (Broadcasts):
Nid|src base  mask    u|l n N   a
(0) 00000000 FFFFFFFF  |    - 00000000
(1) 690F0010 FFFFFFF0 u|  n - 690F0003
(2) 00000000 FFFFFFFF  |    - 00000000
(3) 00000000 FFFFFFFF  |    - 00000000
(4) 00000000 FFFFFFFF  |    - 00000000
```

type **1 Enter**

```
Multicast routing (Broadcasts):
(s)ource base addr:690F0010  (m)ask:FFFFFFF0
(u)ser output:ON  (l)ink output:OFF
(n)et output:ON (N)ode ID:255
next (a)dr:690F0003
```

(s)ource base addr: 690F0010 — Base address compared to the source address of the received packet

(m)ask: FFFFFFF0 — Only masked parts of the `(s)ource base addr` and the packet source address are compared

(u)ser output: ON — Switch for sending a packet copy to the user output of the node

(l)ink output: OFF — Switch for sending a packet copy to the retranslation link output of the node

(n)et output: ON — Switch for sending a packet copy to the retranslation net output of the node

(N)ode ID: 255 — Switch for sending a packet copy to the next node within the same CU (No.0 to No.4, whilst 255 means no next node).

next (a)dr: 690F0003 — When a broadcast packet is sent to the retranslation net output, this address denotes the node from which the acknowledgement will be expected (this is used for secured transfer of a broadcast packet between repeaters on an RF channel). `Next(a)dr:0` means no acknowledgement will be expected (unsecured transmission)

8. SC Channels

The Serial Communication Channels menu enables the setup of all SCC parameters available, ranging from physical layer parameters to the network interface ones. It is divided into CNI, Protocol and Extensions submenus.

8.1. SC Channel to Node Interface

The CNI menu is identical for all channel types. For CNI menu description see chapter Chapter 25, *CNI-channel to node interface*.

8.2. SC Protocol

This submenu contains both low-level channel parameters (bit rate, handshaking, etc.) and access module configuration (type of protocol supported and its parameters).

From MORSE main menu type **SPe Enter Enter**:

```
SCCs:
    n     m     g    b   p8   i   s  XRC D G    o
(0)RS232 ASYNC  SW  19200N81  5 1600 – D 0 MARS-A
(1)RS232 ASYNC  SW  19200N81  5 1600 – D 0 MARS-A
(2)RS232 ASYNC  SW 115200N81  5 1600 – D 0 ASYNC LINK
```

type **1 Enter**

```
Serial Communication Channels:
i(n)t:RS232  (m)ode:ASYNC   dia(g):SW
(b)it/s:19200  (p)ar:NONE  (8)bit:ON  s(T)op:OFF  fr(A)gs:7+9/16
RX (i)dle:5 RX buf (s)ize:1600
TX idl(e):OFF
Handshake: (X)on/Xoff:OFF  (R)TS:OFF  (C)TS:OFF
C(D):ON  (G):0000
pr(o)tocol MARS-A parame(t)ers
ext clocks t(x):OFF  (r)x:OFF sync (w)ord:7E7E
SC(M):0131
```

i(n)t: — the interface type

- (2)RS232 – the American version of the V.24 standard, for connector pin layout see www.racom.eu, Support, Radio modems, MR25, Manual

Note

For other interface use external converter OPI 422 or OPI 485.

(m)ode: — link layer low level mode (part of SCM), it is automatically derived from the selected protocol

- `(a) ASYNC`

- `(b) BISYNC`
- `(h) HDLC`
- `(v) V.110`

dia(g): — diagnostic mode (part of SCM)

(n)NORM — normal (not used, RTS/CTS signals are controlled by the hardware)

(l)LOOP — signals transmitted on the channel from the processor are sent back immediately

(e)ECHO — signal arriving at the port from external equipment is received, and immediately echoed back

(s)SW — standard configuration, the handshake is controlled by the software, including RTS/CTS

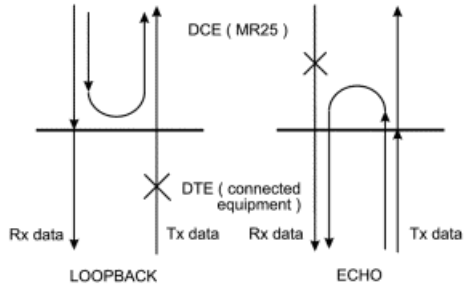

Fig. 8.1: Signal way through interface

- LOOPBACK – Echoed inside CU firmware, thus before RS232 or RS422 converters.
- ECHO – Echoed inside CU firmware. Signal goes through RS232 or RS422 converters to the processor and back.

(b)it/s:19200 — transmission speed in bits per second, to be written in hundreds (type b1152 to get 115200, or b6 to get 600)

(p)ar: — parity setting (part of SCM)

- `(n)NONE` — without parity checking
- `(e)EVEN` — even parity
- `(o)ODD` — odd parity

(8)bit:ON — number of data bits (part of SCM)

- ON — 8 bit
- OFF — 7 bit

s(T)op:ON — the stop bit length consists of the optional part `s(T)op` and the constant part with the adjustable length `fr(A)gs`. The part `s(T)op` is present upon selecting ON and omitted at `OFF`.

fr(A)gs:7+9/16 — this part of the stop bit is present permanently. It's length is adjustable by exchanging the number 7 by 0 to 7, the resulting length is from 9/16 to 16/16 period.

The total length of stop bit can be set using both parameters s(T)op and fr(A)gs in interval 0.563 to 1.000 or 1.563 to 2.000 bits.

RX (i)dle:5 — number of idle characters received from connected equipment before the RX buffer is closed. The lowest possible setting is recommended (5 characters in most cases).

RX buf (s)ize:1600 — the maximum size of the receiving buffer on the relevant SCC port – the possible maximum value is 1600 bytes

TX idl(e):OFF
- `OFF` - no idle transmitted
- `ON` – one idle character is transmitted ahead of every frame

Handshake:

(X)on/Xoff: OFF/ON — ON = handshake performed by transmitting control characters XON and XOFF from both DCE(CU) and DTE

(R)TS: OFF/ON — ON = DCE transmissions are controlled from DTE by RTS

(C)TS: ON/OFF — ON = DCE controls DTE transmissions by CTS

C(D): ON/OFF — sets the initial level of the CD signal

(G):0 — protocol (access module) debug messages level (0 means no debug messages)

pr(o)tocol: — selects the protocol supported by the access module, the list can be changed in new firmware versions:

```
Select new protocol:

(n)-NONE                (B)-ALLEN-BRADLEY DF1
(c)-COMLI               (g)-L&G 870-5-2 IEC
(h)-HIRSH               (H)-HIRSH-D
(y)-HAYES               (G)-GPS
(t)-TRAN SYNC           (X)-EXT SYNC
(M)-MITSUBISHI          (s)-SERVICE
(A)-ASYNC HDLC          (l)-LAP12
(f)-TRANS FRAME LINK    (a)-ASYNC LINK
(b)-ASYNC LINK ++       (e)-MARS-E
(m)-MARS-M              (R)-MARS-U
(N)-MARS-N              (i)-MININET
(T)-MTS                 (o)-REM
(r)-RDS                 (u)-SAUTER
(p)-PR2000              (I)-UNI
```

```
(U)-SURN                (S)-S-BUS

(s)-SCHENCK             (j)-JOHNSON CONTROLS
(z)-ZAT                 (E)-ECS
(R)-RTU                 (M)-MODBUS
(I)-DIRAS               (P)-PPP
(L)-SLIP                (D)-DCF
(p)-PROFIBUS            (F)-FBII
(S)-SKAO                (o)-SKAO-MODBUS
(d)-DEBUG               (r)-RADOS
(e)-RP570               (t)-PROMOTIC
(c)-MELSEC C24          (a)-PARK AIR

(a)-MARS-A              (r)-RLA
(s)-SMS                 (m)-MDU
(i)-INCA                (t)-ARITECH
(S)-SEP                 (A)-Axima
(f)-Sifas               (b)-SEAB
(g)-AEG                 (d)-DACP
(w)-WSR                 (F)-AGFA
(p)-APE                 (n)-ABSONET
(C)-COTAG
```

parame(t)ers — access module parameters (a different set for each respective protocol), see www.racom.eu, Support, Interface Protocols[1]

ext clocks: OFF — switching internal/external clock at RXC pin (for synchronous modes only)

(r)x: OFF — switching internal/external clock at TXC pin

sync (W)ord: 7E7E — sync word (for some of synchronous modes)

SC(M): 0171 — the SCM register content, for SCC special settings, not used during normal configurations

8.3. SC eXtensions

From MORSE main menu type **SXe Enter Enter**:

Extended parameters for SCC. Supplementary parameters for some protocols.

After entering the **SXe** menu the state of registers must be read using the **r Enter** command. When the editing changes are made, save them using the **w Enter** command. The **S,I** commands are not used here.

[1] https://www.racom.eu/eng/support/protocol.html

9. RF Channels

The Radio frequency channel menu enables the setup of interface and protocol parameters independently for each RFC and common properties for the RF modem as a whole.

9.1. RF Channel to Node Interface

The CNI menu is identical for all channel types. For CNI menu description see chapter Chapter 25, *CNI-channel to node interface*

9.2. RF Access

Medium access layer and link layer parameters together with mobile mode parameters are set here. It replaces the old menu RF Protocol (FPe) since version 9.0.0.0.

From MORSE main menu type **FAe Enter Enter** :

```
Rfc access params:
  Access (M)ode:HCSMA t(Y)pe:NORMAL (h)is TO:30 (H)i/Lo rate:2
  Qf(U)ll thr:10 hip(R)q cnt:5
High priority:
  (d)el:16000us (l):16000us (n)um:4 (T)O:10s
  ACK TO=f(i)x:600ms+[0..3]*(v)ar:400ms
  r(e)p:5 (P)rog:OFF
  coding (m)ode:REP (t)ype:DBL
Low priority:
  (D)el:16000us (L):16000us n(u)m:4 T(O):10s
  ACK TO=fi(x):600ms+[0..3]*v(a)r:400ms
  re(p):5 Pro(g):OFF
  coding m(o)de:REP t(y)pe:DBL
de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
```

The packet coming from the node to RF channel is not transmitted immediately. The radio channel access procedure detects, when the RF channel is free and only now is the packet sent. Meantime the packet waits in one of ten queues. One queue for each toa address is occupied. Part of the queues is used for the high priority channel, part for low priority. The packets are classified in the queues according it's handicap bit. It is assigned to each individual packet when coming from user channel to the node, see the menu CNI. The transfer of this bit at the retransmission is in the development. At present (version 10.0.28.0) it is not transferred. The packet are also in the next radio steps putted in the queues with the high priority.

This principle is applied for CU MR25 from fw 9.0.32.0 with limited extent of queues (number of queues 5, depth max. 5).

RFC access params

Common parameters for both priority levels:

(M):HCSMA Access (M)ode:HCSMA - switch-on the HSCMA mode

- (h) HCSMA - ON
- (n) NONE - OFF

(Y):NORMAL t(Y)pe:NORMAL - Type of access to the RF channel

- `NORMAL` — normal
- `R-CTRL` — receiver-controlled access mode CU sends a REQ packet (the same length as the ACK packet) to the counterpart at first. It contains an announcement of an intention to deliver a data packet of some length. The opposite CU then transmits a "reservation" packet (RES) with information for all CUs on the channel, that the channel is reserved for given time for given address only.
- `RXonly` — the channel must not transmit

(h):30 (h)is TO:30 - time dimensions (sec)of list of received packets, which is used for discarding of repeated packets

(H):2 (H)i/Lo rate:2 - the rate of packet transmitting frequency from the high priority queue and from the low priority queue. The high priority packets are transmitted here twice more often.

(U):10 Qf(U)ll thr:10 - the special packetlost is generated after reaching this number of packet in the queue. The packet continues to be accepted until the maximal queue length 10 packets.

(R):5 hip(R)q cnt:5 - the number of high priority queues. Next 10-R-1=4 queues have the low priority, one queue is the collecting. It collects packets having the toa address, to which is not available a free queue. This queue collects the packets without regard to the packet priority level.

High priority

The parameters for packet having the high priority. The analogical parameters follows for the low priority queues.

Access

The time in which the radio tests the RF channel consists of a fixed part (fixed access period) and a random part (slot length multiplied by a random number), access period=d+r×l, where r comes from the interval 0 to (n-1). The fixed part (typically 16 ms) allows the ACK to pass without interference.

`(d):16000` (d)el:16000us - Fixed part of access period in microseconds

`(l):16000` (l):16000us - Slot length in microseconds

`(n):4` (n)um:4 - Max. number of slots

When the channel access tunning it is recommended to let the (d)el and (l) parameters on the values 16000 (for 25kHz channel) and change the (n)um only.

`(T):10` (T)O:10s - Access timeout in sec. The time limit for the station to try and get onto the RF channel. If this timeout is exceeded, the modem discards the packet and sends an error message.

ACK

ACK timeout is the period that the RFC protocol waits for acknowledgement from the counterpart. To avoid collision deadlock between stations transmitting long packets and not detecting other's signal (i.e. CSMA principle cannot work), it is recommended to set time values which correspond to the maximum transmission time of the supposed longest packet to both parts of the ACK timeout.

(i):600 ACK TO=f(i)x:600ms - fixed part of ACK timeout in ms

(v):400ms +[0..3]*(v)ar:400ms - variable part of ACK timeout in ms

(e):5 r(e)p:5 - The maximum number of repeated messages on the RF channel (unacknowledged by the counterpart)

(P):OFF (P)rog:OFF - for service purposes

Coding

The data coding in the RF channel

(m) (m)ode:REP - the coding activation

- `NO` - RFC data is not coded
- `REP` - RFC data is coded in repeated transmissions
- `ALL` - RFC data is always coded

(t) (t)ype:DBL - type of coding of RFC data

- `NO` - no coding
- `DBL` - doubled blocks The packet is divided to blocks of 10 bytes each. Every block is secured by its own CRC and all data blocks are then transmitted two times, i.e. the actual size of the transmitted packet is more then two times greater than without coding. The received data is completed from blocks where CRC check is okay.
- `F5 STUFF` - special coding for production testing

9.3. RF Protocol

This menu is replaced by a new one **FAe** (RF Access) since 9.0.0.0 version. The mobile mode parameters are relocated into menu **DGei**.

Medium access layer and link layer parameters together with mobile mode parameters are set here.

From MORSE main menu type **FPe Enter Enter** :

```
RF channels:
    Access            |ACK             |coding |Mobile
 id a     del l num TO|fix var rep P hT|mod typ|base   mask  center  per
(0) NORMAL 16 16  4 10| 600 400  5   30|REP DBL|OFF
(1) NORMAL 16 16  4 10| 600 400  5   30|REP DBL|OFF
(2) NORMAL 16 16  4 10| 600 400  5   30|REP DBL|OFF
(3) NORMAL 16 16  4 10| 600 400  5   30|REP DBL|OFF
(4) NORMAL 16 16  4 10| 600 400  5   30|REP DBL|OFF

de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
```

type **1 Enter**

```
Radio Frequency Channels:
(a)ccess type:NORMAL (d)el:16ms
(l):16ms (n)um:4 (T)O:10s
ACK TO=(f)ix:600ms+[0..3]*(v)ar:400ms
(r)ep:5 (P)rog:OFF (H)ipr:OFF (h)is TO:30
coding (m)ode:REP (t)ype:DBL
(M)obile mode:OFF  (b)ase:00000000  mas(k):00000000
(c)enter addr:00000000  (p)eriod:0sec
```

Access

The time in which the radio tests the RF channel consists of a fixed part (fixed access period) and a random part (slot length multiplied by a random number), access period=d+r×l, where r comes from the interval 0 to (n-1). The fixed part (typically 16 ms) allows the ACK to pass without interference.

`(a)cces type` — Type of access to the RF channel

- `NORMAL` — normal
- `R-CTRL` — receiver-controlled access mode CU sends a REQ packet (the same length as the ACK packet) to the counterpart at first. It contains an announcement of an intention to deliver a data packet of some length. The opposite CU then transmits a "reservation" packet (RES) with information for all CUs on the channel, that the channel is reserved for given time for given address only.
- `RXonly` — the channel must not transmit

`(d)el:16ms` — Fixed part of access period in ms.

`(l):16ms` — Slot length in ms.

`(n)um:4` — Max. number of slots.

When the channel access tunning it is recommended to let the (d)el and (l) parameters on the values 16 (for 25kHz channel) and change the (n)um only. Lower (n)um causes the higher priority and conversely.

`(T)O:10s` — Access timeout in sec. The time limit for the station to try and get onto the RF channel. If this timeout is exceeded, the modem discards the packet and sends an error message.

ACK

ACK timeout is the period that the RFC protocol waits for acknowledgement from the counterpart. To avoid collision deadlock between stations transmitting long packets and not detecting other's signal (i.e. CSMA principle cannot work), it is recommended to set time values which correspond to the maximum transmission time of the supposed longest packet to both parts of the ACK timeout.

`ACK TO=(f)ix :600ms` — fixed part of ACK timeout in ms

`+[0..3]*(v)ar: 400ms` — variable part of ACK timeout in ms

`(r)ep: 5` — The maximum number of repeated messages on the RF channel (unacknowledged by the counterpart)

`(P)rog: OFF` — for service purposes

(H)ipr:OFF — for service purposes

`(h)is TO: 30` — time dimensions (sec)of list of received packets, which is used for discarding of repeated packets

Coding

The data coding in the RF channel

`coding (m)ode:`

- `REP` — RFC data is coded in repeated transmissions
- `NO` — RFC data is not coded
- `ALL` — RFC data is always coded

`(t)ype:` — type of coding of RFC data

- `DBL` – doubled blocks The packet is divided to blocks of 10 bytes each. Every block is secured by its own CRC and all data blocks are then transmitted two times, i.e. the actual size of the transmitted packet is more then two times greater than without coding. The received data is completed from blocks where CRC check is okay.
- `NO` – no coding
- `F5 STUFF` – special coding for production testing

Mobile mode

The mobile station parameters

`(M)obile: OFF/ON` — setting this parameter switches the RFC to mobile mode

`(b)ase: 00000000` — comparative address for choice of base stations

`mas(k): 00000000` — if any station which can be received on the RF 0 channel is to be taken as base, masked parts of the (b)ase parameter and the base station address must be equal

`(c)enter: 00000000` — address of central station, where this mobile has to report to

`(p)eriod: 0sec` — period (sec) for reporting to the central station (0 means no periodical reports are sent)

9.4. RF Modem

The RF modem configuration data (common for all RFCs, as there is only one physical RF channel). It is strongly recommended to use default values here.

From MORSE main menu type **FMe Enter Enter**

```
RF Modem:
RFTX disabl(e):OFF
(m)ode:MORSE
invert (T)X:ON (R)X:ON
TX blocks: (i)dle:0  sy(n)c:3
DQ (t)reshold:5 Valid for SW < 805  (s)ync TO:40
RSS treshol(d):105  RSS m(o)de:RSS treshold
FX(9)19  FX(5)89
(W)dog:0min  (l)oging:OFF
de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
>>
```

`RFTX dis-abl(e): OFF` — `OFF` — normal state

`ON` — RF transmitting disabled

`(m)ode: MORSE` — `MORSE` — MORSE system mode

`C92` — mode C92 (RD300 FS compatible)

`invert (T)X: OFF/ON` — TX and/or RX modulation inverted; use default values when in doubt

`invert (R)X: OFF/ON` — TX and/or RX modulation inverted; use default values when in doubt

`TX blocks: (i)dle:0` — the delay before sync.blocks transmitting

`(s)ync: 3` — number of sync blocks transmitted (6 bytes each)

`(s)ync TO: 40` — frame sync timeout (msec); after detecting the first sync byte the RF modem is waiting for frame synchronisation sequence . When this timeout expires, the Noise

sync counter is incremented (see `(s)ervice (R)X modem (s)tatus noise sync`).

`DQ (t)reshold:5` — DQ limit for access to the channel - RFC does not start the transmitting when the audible signal better then set limit occurs on the RFC, it is replaced by `RSS treshol(d)` parameter since fw 806

MR25 from fw 10.0.50.0 employs DQ treshold only

MR400 from fw 10.0.50.0 employs RSS treshold only

`RSS treshol(d):105` — access to the channel - RFC does not start the transmitting when the audible signal stronger then set limit occurs on the RFC, the RSS treshold is used for the MR400 series since fw 806

`RSS m(o)de:` — for MR400 series only

`(t) RSS treshold` — RSS treshold is active

`(o) off` — function treshold is off

`FX(9)19` — setting of the signal processor for MR25 and MR400, the `(s)peed` parameter indicates the RF channel bandwidth. This bandwidth is a part of the production code[1] since 12/2008

- `(s)peed:1/1` — 25kHz
- `(s)peed:1/2` — 12.5kHz

`FX(5)89` — setting of the signal processor for MR900, the parameter `(s)peed` indicates the bandwidth of the RF channel

- `(s)peed:1/2` — 125kHz

`(W)dog:` — for service purposes

`(l)oging:` — for service purposes

[1] http://www.racom.eu/eng/download/morsecode.html

10. Ethernet

Note

Apply for software version 505 and above.

10.1. Ethernet Channel to Node Interface

The CNI (Channel to Node Interface) menu is identical for all types of channels. It is described in chapter Chapter 25, *CNI-channel to node interface*

Let's note that the IP-M-IP and Morse Application Server communication use the user output but M-IP-M uses the retranslation output, see later.

The external form of the menu:

```
Channel to Node Interface:
    retranslation     |   user                      lim
 id N  A t          m | N  A t Base     m  sec brc S   e
(0) 0   NO AR         | 1 MASK 00000000/08 ON  OFF NONE

de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
```

10.2. Hardware configuration

From MORSE main menu type **Ehe Enter Enter**

```
Ethernet:
(e)nable:ON
(s)peed:auto bps
(p)romiscuous mode:OFF
d(u)plex:auto
(T)imer init level:0s
Eth. adr(H)i:0002h
Eth. adr(M)i:A953h
Eth. adr(L)o:73B7h
(R)XQ depth:16
R(X)Q count:16
(z)Status phy
Restart (i)nit level:FEC
(d)sc: 5810
de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
>>
```

The default values are recommended in this menu. The others are suitable for service or development purposes.

Meaning of items:

Ethernet (e)nable
- `ON` – Ethernet module is active
- `OFF` – Ethernet module is inactive

(s)peed:auto bps — choice of ethernet speed
- `(0)` auto bps — default
- `(1)` 10Mbps
- `(2)` 100Mbps
- `(3)` auto bps, restart on Init

(p)romiscuous mode
- `OFF` – accepts the packets addressed to its own address only
- `ON` – accepts all packets

Full (d)uplex
- `(a)` auto — default
- `(f) full` — duplex traffic, connection by RJ45 cable
- `(w) full with flow control`
- `(h) half` — simplex traffic, connection by coaxial cable

(T)imer:0s — the Ethernet module test packets sending interval
- `0` — off
- `1` — in 1sec period

Eth. adr(H)i: 0002h — high part of module address in Ethernet network

Eth. adr(M)i: A958h — middle part of address

Eth. adr(L)o: 41E9h — low part of address

(R)XQ depth: 16 — the depth of RXQ queue for packet transfer from the Ethernet module into CU, selectable from 1 to 32

R(X)Q count: 16 — number of buffers which are used by Ethernet module during operation, selectable from 1 to 32

(z)Status phy — for development purposes

10.3. Setting protocol parameters

From MORSE main menu type **EPe Enter Enter**.

```
  Internet Protocol:
Eid| ip address            |    net mask           |          gw       |
(0) C0A80009 192.168.0.9    FFFFFF00 255.255.255.0   00000000 0.0.0.0

de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
>>
```

Continue by: **0 Enter**

```
Etp:
(i)p adr:C0A80009h
(g)ateway adr:00000000h
(n)et mask:FFFFFF00h
(G):0000
(d)sc: 6029
AR(P) parameters
parame(t)ers
de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
>>
```

Meaning of items:

(i)p adr: C0A80009h — address of Ethernet module in IP network

(g)ateway adr: 00000000h — IP address of gateway in foreign networks

(n)et mask: FFFFFF00h — mask defined address space in own IP network

(G): 0000 – debugging level (0–16)

AR(P) parameters – parameters for ARP communication:

Type **P Enter**:

```
ARP:
(A)RP ttl:30s
A(R)P timeout:50ms
proxy arp (m)ode:NORMAL
Proxy Arp (B)ase:00000010
Proxy Arp (M)ask:000000F0
(p)rint ARP table
```

```
(c)lean ARP table
(s)ave ARP table
(r)estore ARP table
ipg(W) restart
(q)uit
>>
```

(A)RP ttl: 30s — time of life ARP record

A(R)P timeout: 50ms — timeout for ARP answer

proxy arp (m)ode — next three parameters define, to which ARP request should Ethernet channel send the answer

WARNING — the setting `(p)` or `(e)` causes certain danger for IP network in case of wrong setting `(B)` or `(M)`!

- `(n) NORMAL` - ARP report is generated only for addresses without `(n)et mask` in relation to `(i)p adr`, e.g. for `C0A8BB01`
- `(p) POSITIVE` - ARP report is generated only for addresses within `(n)et mask` and within `Proxy Arp (M)ask` in relation to `Proxy Arp (B)ase`, e.g. for `C0A80011`
- `(e) NEGATIVE` - ARP report is generated only for addresses within `(n)et mask` and without `Proxy Arp (M)ask` in relation to `Proxy Arp (B)ase`, e.g. for `C0A80001`
- `(N) NORMAL LOCK` - ARP table is locked, see the description locking ARP
- `(P) POSITIVE LOCK` - ARP table is locked in the POSITIVE mode
- `(E) NEGATIVE LOCK` - ARP table is locked in the NEGATIVE mode

See the example in the MORSE Guide 3[1].

(p)rint ARP — displays the current ARP table

(c)lean ARP — erases the ARP table

(s)ave ARP — saves the ARP table into NVHeap memory

(r)estore ARP — reads the ARP table from NVHeap memory

ipg(W) restart — restarts the ipgw protocol

Locking of ARP table (from fw 10.0.94.0)

The ARP table changes dynamically. After exchange ARP REQ and ARP REP frames the new item is added in the table or the time expiration record is updated. The record is deleted when the lifetime is expired. This dynamics can be switched off by locking the table using the command `proxy arp`

[1] https://www.racom.eu/eng/support/morse-m3/eth-ip-m-ip.html

`(m)ode:...LOCK`. Use the analogical proxy arp mode, e.g. for NORMAL use NORMAL LOCK. Then the ethernet protocol communicates only with IP addresses contained in the locked ARP table. It answers these ARP REQ only which are contained in the table. The old records are not cleared from the ARP table. Thus prevents the frame transfer from unknown IP addresses. The locking process:

- Configure normally the ethernet channel, e.g. in mode NORMAL
- Check up, if all needed items in ARP table are created. Use command `(p)rint ARP table`.
- Save the table in NVHeap memory using command `EPe 0Ps`. The state of ARP table at the saving moment is crucial for next traffic in the state LOCK.
- Backup the NVHeap entry into Flash memory using command `cb` i.e. `(c)onfigure (b)ackup`.
- Lock the ARP table by command e.g. NORMAL LOCK `EPe 0PmN`, `(q)uit`, `(I)nit`, `(W)rite`. Thereby the entry from NVHeap memory is copied into ARP table.
- Check the content of the locked table ARP by command `EPe 0Pp`, the table have to contain the records for all participiants of the expected communication. The zero is on the place of the record living time.
- This procedure accomplish as necessary on others CUs on the MORSE-IP interface.

Others operations with the table:

- The table must not be locked earlier then it is saved by command `(s)ave ARP table`. Or else the ethernet communication is blocked immediately or after CU restart.
- The reading of saved table occurs at the initialization of `proxy arp (m)ode:LOCK`.
- If the CU starts with exhausted back-up battery, then the configuration (include ARP table) is restored from the flash memory. That's why the backup `cb` should be done.
- The state of saved table in NVHeap can be checked in the menu `EPe 0P` using commands `(r)estore ART table` and `(p)rint ART table`.

The locked ARP table behaviour:

- The timeout are switched out and the records are not deleted.
- The ethernet channel does not send the ARP when transmits the frames in the ethernet network. The frames can be send to IP addresses contained in the table only.
- The receiving of frames from ethernet network is possible from these addresses only, which corresponds with IP and MAC written in the table. The others are cancelled.
- Protocol answers this requests ARP REQ only, whose IP and MAC are in the table.
- Protocol does not written a new records in the table.
- If the hw part is exchanged at some participant, then the communication with him stops (different MAC address).

The transparent illustration of the processes described above:

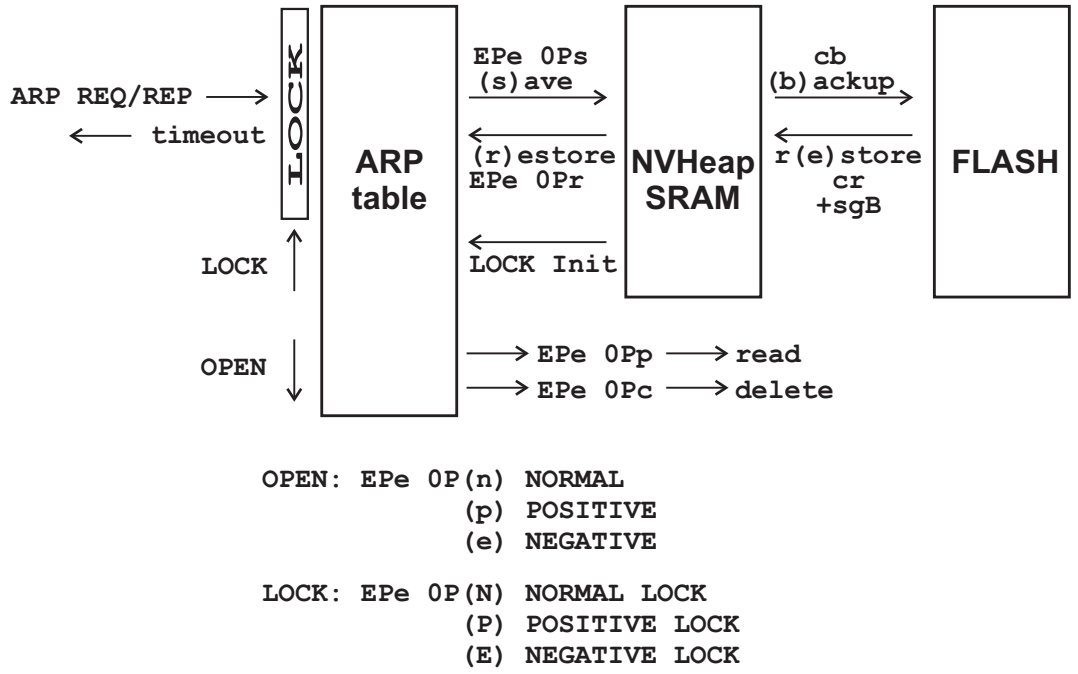

Fig. 10.1: ARP table

An example of current (unlocked) ARP table :

```
ARP cache:  IP          Ether            time
00 192.168.068.017    0010:6076:8351        19
01 192.168.068.016    0021:7093:d178        22
```

The record contains the IP and Ethernet address of the station, with which the ARP packet was exchanged and the remaining validity time for this record, see `(A)RP ttl`

An example of locked ARP table :

```
ARP cache:  IP          Ether            time
00 192.168.068.017    0010:6076:8351        0
01 192.168.068.016    0021:7093:d178        0
```

Note

The Setr 10.0.94.0 or later is necessary for ARP LOCK mode settings.

The continuation of EPe 0 menu by the last item:

parame(t)ers — setting of other parameters

Type **t Enter**

```
  INTERNET PROTOCOL GATEWAY:
M-IP-M:
  (A)rt:0; write (E)nable:ON
  (b)ase:00000000  MAS(K):00000000  s(h)ift:0000 ->set Security off!
```

```
  (r)epeats:0000  Sec(u)rity:ON  (t)imeout:0  (p)roxy timeout:0s
  (f)rag size:400bytes  (g)lue (append) up to:0packets
IP-M-IP:
  (I)PArt:0
  b(a)se:00000000  (m)ask:00000000
  IP(F)rag. size:552
MAS:
  (s)Art:0; write (e)nable:ON
  (B)ase:00000000  (M)ask:00000000
     BEWARE! IF YOU CHANGE CONTENT OF THE ART TABLE,
     YOU SHOULD RESTART (INIT) THIS PROTOCOL!
(P)inger (S)tatus
Firewall (T)ype:0000 A(d)dress:00000000 Mas(k):00000000
Unix Time:
  e(n)able:OFF  per(i)od:0s
  time (H)ost:00000000
(q)uit
>>
```

Before the gateway description let see the commands **(P)inger** and **(S)tatus** :

```
ICMP ping:
(t)arget:00000000h 0.0.0.0
pe(r)iod:1000ms
p(a)ttern:DEDAh
si(z)e:100
(s)tart r(e)port sto(p)
(q)uit
>>
```

Using the **(s)tart** command you can ping the ICMP packet to the IP address `(t)arget` and in this way test the communication.

(t)arget: C0A80001h 192.168.0.1 — the target IP address

pe(r)iod: 5000 — the repeat period [ms]

p(a)ttern: DEDAh — the sended hexa characters

si(z)e: 100 — the whole length of packet sended

(s)tart — start packet sending

r(e)port — the sending and receiving report

sto(p) — stop packet sending

IP frames multicast

The menu Firewall allows process the IP frames multicast:

```
Firewall (T)ype:0000 A(d)dress:00000000 Mas(k):00000000
```

(T)ype
- 0 - OFF — off
- 1 - TELENOR, 2 -ABB — customers applications
- 3 - mcast — IP frames multicast are received and sent to the MORSE network as a broadcast packets

Some received IP frames are spread as a broadcast packets in the mode `3 - mcast`. Those IP frames are choose according to their `IP dest` address by means of the parameters `A(d)dress` and `Mas(k)`, e.g.:

```
A(d)dress:C0A821FF
Mas(k):FFFFFFFF
```

or

```
A(d)dress:000021FF
Mas(k):0000FFFF
```

The outgoing CU must have set the parameter `(T)ype:3`, the parameters `A(d)dress` and `Mas(k)` are empty, see the example in the MORSE Guide 3[2]. All CUs on the MORSE path shall have configured properly the parameters in the Broadcast menu.

The statements `(S)tatus` group are used for service purposes.

The `Unix Time` group enables getting time from the Unix machine.

We can show the related menus again for better orientation including Art tables.

[2] https://www.racom.eu/eng/support/morse-m3/eth-ip-m-ip.html

```
Etp:
(i)p adr:C0A80009h
(g)ateway adr:00000000h
(n)et mask:FFFFFF00h
(G):0000
AR(P) parameters
parame(t)ers
...
```

```
  INTERNET PROTOCOL GATEWAY:
M-IP-M:
  (A)rt:0; write (E)nable:ON
  (b)ase:00000000  MAS(K):00000000  s(h)ift:0000 ->set Security off!
  (r)epeats:0000  Sec(u)rity:ON  (t)imeout:0  (p)roxy timeout:0s
  (f)rag size:400bytes  (g)lue (append) up to:0packets
IP-M-IP:
  (I)PArt:0
  b(a)se:00000000  (m)ask:00000000
  IP(F)rag. size:552
MAS:
  (s)Art:0; write (e)nable:ON
  (B)ase:00000000  (M)ask:00000000
     BEWARE! IF YOU CHANGE CONTENT OF THE ART TABLE,
     YOU SHOULD RESTART (INIT) THIS PROTOCOL!
(P)inger (S)tatus
Firewall (T)ype:0000 A(d)dress:00000000 Mas(k):00000000
...
```

```
ART No 1:
items: 3
default gw: C0A80F00
dest:    gw:
6990508E C0A80F05
69905091 C0A80F06
69905094 C0A80F07

ART No 2:
items: 2
default gw: 693A0000
dest:    gw:
C0A8050E 6932000E
C0A80506 693A000D

ART No 3:
items: 3
default gw: 00000000
dest:    gw:
690F9002 22B8711D
690F9042 22B85024
690F9043 22B80A01
```

10.3.1. M – I P – M

Packet retranslation from the MORSE network through the IP network to the MORSE network again.

Configuration

- the Ethernet channel is connected to the node net output
- the node is connected to the retranslation output from Ethernet channel
- nonzero (1-4) content of (A)rt activates the M-IP-M mode

Processing of the packet:

1. The packet with MORSE address `TO` comes from the MORSE network.
2. Mask conversion of addresses

   The MORSE address `TO` is compared with the parameter `(b)ase` in those parts where the `MAS(K)` has zeroes (in bits). If there is a match the address `IPdest` is created by masking.

   The MORSE address `TO` is compared with the parameter `(b)ase` in those parts where the `MAS(K)` has zeroes (in bits). If there is a match the address `IPdest` is created by masking.

   For bits where there are ones in the `MAS(K)` there must be zeroes in the `(b)ase`.

   For mask conversion of an address `Sec(u)rity:OFF` is required.
3. Conversion of addresses via table Art

   If the conditions for mask conversion of an address were not complied with then table `Art` is used.

   In the `(A)rt` table pertaining to M-IP-M mode the IPdest address is determined according to the MORSE address `TO`.

   If the `TO` address isn't in the table then `default gw` is used from this table.
4. The next direction in the IP network is determined according to destination IP address with help of parameters from the first part of menu `Etp`. If the condition

   ```
   (IPdest & (n)et mask) = ((i)p adr & (n)et mask)
   ```

   is fulfilled then the packet is sent to the destination IP address. If the condition is not fulfilled then the packet is sent to the address `(g)ateway adr`.
5. The MORSE packet is completed with the IP header and is sent to the IP network. The following parameters can be used:

   (r)epeats: 0005 — max. number of repeats when sending through IP

   (t)imeout: 100 — timeout (ms) for waiting for ACK in IP

   (p)roxy timeout: 0 s
   - 10 and more - the interval of sending identification packets through the proxy server

- 0 - off

(f)rag size: 400 bytes — minimal adjustable value is 200, recommended 400 to 1400 – max. length of packets for IP, longer ones will be divided

(g)lue (append) up to: 0 packets — in progress, max. number of joinable packets, put in the value 0

6. The packets travels according to IP network rules.

7. Upon output from the IP network the IP header is removed and the packet is handed to the MORSE network and continues on its way according to MORSE routing.

Note

By using the `write(E)nable:OFF` parameter we can disable the automatically fulfilling of Art table, which comes up in some modes.

Conversion of addresses **using a mask** with `Sec(u)rity:OFF` permits an unlimited number of addresses to be used and makes better use of the transmission capacity of the IP channel. Security, if required, needs to be implemented on different communication layers. Fragmentation is not available here. The recommended value is `(f)rag size: 1400 bytes`, which is the max. size of transmitted packet.

Conversion of addresses **using table Art** is limited to approx. 100 addresses. Only a small portion of the transmission capacity of the IP channel is used. On selecting `Sec(u)rity:ON` fragmentation proceeds in parts according to parameter `(f)rag size: 1400 bytes`. On selecting `Sec(u)rity:OFF` fragmentation is not available. The recommended value is `(f)rag size: 1400 bytes`, which is the max. size of transmitted packet.

10.3.2. IP – M – IP

Retranslation of packets from the IP network through the MORSE network to IP network again.

Configuration

- Eth is connected to the node user output
- node is connected to the Eth user output
- nonzero (1-4) content of `(I)PArt` activates the IP-M-IP mode

Processing of the packet runs in these steps:

1. The packet with destination address `IPdest` comes from the IP network to my address `MyIP`.
2. The MORSE address is found in next steps:

Art

The Morse address `gw` is found in table `(I)PArt` according to the `IPdest` address, which signifies the destination address in the MORSE network.

Mask

If the IPdest is not found in the left column of (I)PArt, then comparison of the addresses (b)ase from IP-M-IP menu and IPdest is made by application of the bit-inverted value of (m)ask, thus

```
((b)ase & ~(m)ask) = (IPdest & ~(m)ask)
```

If this condition is fulfilled, then the destination MORSE address Morsedst is determined as my Morse address MyMorse masked by the bit-inverted value (m)ask completed by IPdst masked by (m)ask thus

```
Morsedst = (MyMorse & ~(m)ask) | (IPdst & (m)ask)
```

Example 1:

```
MyIP       192.168.15.128  C0 A8 0F 80
IPdst      192.168.15.04   C0 A8 0F 04
(m)ask     0xFF            00 00 00 FF
           condition fulfilled
MyMorse                    69 0F 90 43
Morsedst >>                69 0F 90 04
```

Default gw

If the condition is not fulfilled or if (m)ask = 00000000, then the address default gw is used from (I)PArt table.

Example 2:

```
MyIP       192.168.15.128  C0 A8 0F 80
IPdst      192.168.05.06   C0 A8 05 06
(m)ask     0xFF            00 00 00 FF
           condition is not fulfilled
ART1 default gw            69 3A 00 00
Morsedst >>                69 3A 00 00
```

3. IP packet is completed by the MORSE header and is sent in MORSE network to the destination address.
4. The packet goes through the network according to MORSE routing.
5. At the output point from the MORSE network in the CU, which has Morse address Morsedst and simultaneously IP address CUadrIP the MORSE head is removed.
6. The next routing in the IP network is defined according to the destination IP address with use of the parameters from the first part of menu Etp. If the condition

   ```
   ( IPdest & (n)et mask ) = ( CUadrIP & (n)et mask )
   ```

   is fulfilled then the packet is sent to the destination IP address.

   Example 3:

```
CUadrIP                          C0 A8 0F 20
packet's IPdst                   C0 A8 0F 04
(n)et mask                       FF FF FF 00
                 condition
                 fulfilled
packet send to IP                C0 A8 0F 04
```

If the condition is not fulfilled, then the packet is sent to the Ethernet address which has the equipment with IP address `(g)ateway adr`.

Example 4:

```
CUadrIP                          C0 A8 0F 20
packet's IPdst                   C0 A8 10 04
(n)et mask                       FF FF FF 00
                 condition is
                 not fulfilled
(g)ateway adr                    C0 A8 0F 00
packet send to IP                C0 A8 0F 00
```

10.3.3. MAS

It is used for packet sending to and from the MORSE network through the MORSE Application Server into the IP network.

Configuration

- the Ethernet channel is connected to the node user output
- the node is connected to the Ethernet user output
- parameter `(s)Art` different then zero or `off` (from fw 10.0.18.0) activates the MAS mode

The packet sending from MORSE network through MAS to the IP network.

1. The packet with destination MORSE address `MasMorse`, which belongs to the application server, comes from the MORSE network.
2. Check as to whether it fulfils the condition

   `(MasMorse & ~(M)ask) = ((B)ase & ~(M)ask)`

   If yes then the packet is sent to the Morse Application Server, if not then it is sent to `IP_M_IP`. The next condition for MAppS is the user packet type (`user_data 0x09, prot_data 0x0A`), not the retranslation.
3. In the table `(s)Art` belonging to Morse App. Server the item `gw` is found according to the destination MORSE address and then it is processed as follows:
   - The lower part of `gw` is used as the lower part of resulting IP address of length 14 bits from the right. The 15th and 16th bit are used for internal purposes. The upper part of the IP address (18 bits) is completed from its own CU IP address.

- The higher part of `gw` (as a rule `22B8 hex`) is converted into (dec), here 8888 , and becomes a UDP port number.

4. The packet is sent to the IP address and UDP port created in this way.

5. So the packet coming from MORSE network to the MAS (Morse Appl. Server) contains MORSE addresses `src` and `dst`. After processing in MAS it contains the IP address, UDP port number and `src` MORSE addr.

Note

By using the `write(e)nable:OFF` parameter we can disable the automatically fulfilling of Art table, which comes up in some modes.

The packet sending from IP application through MAS to the MORSE network

1. The packet with destination MORSE address `MorseDst` comes from the application (daemon) running on the address `IPAddr` and port `AppPort`.

2. The items `AppPort` and `IPAddr` are assembled to create the `gw` , which is found in the Art table. The 14 LSB from IPAddr creates the right part of `gw` , the bit No 15 is 1 for broadcast, the bit No 16 is 0 for manually created Art, the bits No 17 to 32 are borrowed from `AppPort`.

3. The created item `gw` is used to found the respective item `dest` in the Art table. The founded item `dest` becomes the `source` address in the MORSE packet, which is sent into MORSE network.

4. So if the application sends the report back through MAS to the MORSE network, then it contains the IP address of the application, UDP port number and `dst` MORSE address. After processing in MAS it contains MORSE addresses `src` and `dst`.

The default MORSE address for IP —> MORSE direction.

Works in sw 570 and later.

The packet sent from the Application via MAS to the MORSE network in some cases has not assigned the port number. It is possible to define the default MORSE address for the Application, which is found in the Art table in three steps:

1. After unseccesful searching in the `gw` collumn in Art table

2. is found the address `00000000` in the `dest` collumn. The item `gw` belongs to this address, e.g. `22B86420`, where `22B8` is the choosen standard port number and `6420` is the low part of Application IP address. This pair

   `00000000 22B86420` must be located on the end of Art table, so the pair mentioned in next point e.g.: `744901BB 22B86420` appears above it.

3. This item `22B86420` is then found in the `gw` collumn and the item `dest 744901BB` belonging to it becomes the MORSE address `source` in the sent packet.

Art table choice in MAS mode

In older fw the Art table is selected by writing number 1 to 4 into parameter `(s)Art:`, the value 0 switches off the MAS mode.

From fw 10.0.18.0 choose from the menu:

```
    sArt
(f) off             ...MAS mode turned off
(1) ART1            ...Art1
(2) ART2            ...Art2
(3) ART3            ...Art3
(4) ART4            ...Art4
(9) ART9            ...MRouter writes into file instead of the Art table
(s) single address  ...only one IP address connected, see bellow
```

Single address mode is determined for the communication with only one IP address which is greater by one than the address written in menu EPe. The communication runs between ports 8888dec and 8888dec. Configuration example:

```
  Internet Protocol:
Eid| ip address             |    net mask            |        gw        |
(0) C0A801E7 192.168.1.231   FFFFFF00 255.255.255.0   00000000 0.0.0.0

MAS:
  (s)Art:single address; write (e)nable:ON
  (B)ase:690F5600  (M)ask:00000000
```

PC having address 192.168.1.232 is connected by the command:

```
setr.exe -pIP192.168.1.231 -pw690f5600 -pm8888
```

The choice single address is suitable for communication PC-CU using the protocol SLIP.

The format of the frames of MAS and UDP is described in the article Format of UDP datagram IPGW for Morse[3]

[3] https://www.racom.eu/eng/support/morse-m3/eth-ipgw.html

11. NAG Channels

The Network Agent Channels behaves similarly like Serial channels but they does not have the physical inputs/outputs. The packets after processing in NAG are sent into node in the same CU. The menu is divided into two sections, CNI and Protocol submenus.

11.1. NAG Channel to Node Interface

The CNI menu is identical for all channel types. For CNI menu description see chapter Chapter 25, *CNI-channel to node interface*

11.2. NAG Protocol

The NAG can run in increasing number of modes. The content of menu differs for individual modes. The setting for TMM (Tunnel Morse-Morse) mode is shown here.

From MORSE main menu type **GPe Enter Enter**

```
Nag:
 id |typ|subt|  addr |  mask  |
(0) 0001 0000 00001000 FFFF0FFF
(1) 0000 0000 00000000 00000000

de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
>>
```

type **0 Enter**

```
Net Agents:  (t)ype:TMM
(p)arameters
(I)nit (W)rite
(q)uit
>>
```

type **t Enter**

```
    type
(n) NONE
(t) TMM
(r) RTU
(d) ADIO
(k) KATO
(p) TSTP
```

choose **t Enter** for type Tunnel Morse-Morse

type **q Enter**

choose **p Enter** for parameter setting

```
Network agent parameters:
(t)mm
(r)tu
a(d)io
(k)ato
```

choose **t Enter** again for TMM parameters

```
Tunnel Morse-Morse parameters:
(s)ubt:0000
(a)ddr:00001000
(m)ask:FFFF0FFF
(q)uit
>>
```

The retranslation packet incoming in Tunnel Morse-Morse is packed in new service packet (see MORSE Guide) and the new destination address is created for it. This address is derived from original `dst` address by means of next two parameters.

(m)ask: FFFF0FFF — Where there are "1" bits in this mask the original `dst` address is maintained.

(a)ddr: 00001000 — In those bits where the mask is zero the `dst` address bits are replaced with the respective values from the parameter `(a)ddr`.

(s)ubt: 0000 — Subtype is not used in TMM mode.

Description of individual Net Agents are in the MORSE Guide 2[1] or in chapter Interface Protocols[2].

[1] https://www.racom.eu/eng/support/morse-m2/tunnel.html
[2] https://www.racom.eu/eng/support/protocol.html

12. Address resolution table

The `(A)rt` menu is used for creation and editing of Address resolution tables. These tables are used on Channel-to-Node interface, when a complex conversion of the user address space to the MORSE address space is needed. Ethernet and some protocol access modules can use these tables, too. The `(A)rt` menu allows for editing table items regardless of the purpose to which the table is used.

The Art Tables extent is limited. The tables Art1 to Art4 together can contain at the most 252 items. It is recommended use less then a 100 items in one table. The more space offers fw Walrus in hw M-Server in case of mode MAS.

The Art table consists of needed amount pairs `(d)est` + `g(w)` length 4 + 4 byte. The `(d)est` is e.g. MORSE address and `g(w)` is IP address. The MORSE addresses are converted to IP addresses by the table. The content of pairs can be different according to needs. Example:

```
690F0041 C0A80120
690F0042 C0A80125
690F0072 C0A80114
690F1589 C0A80137
```

From MORSE main menu type **A Enter**

```
Address Resolution Table: (N)o:1
(d)est:00000000 g(w):00000000
(g)et p(u)t (D)elete item
de(f)ault gw
(c)lear
(r)ead all
(q)uit
>>
```

(N)o :1 — number of table edited (1, 2, 3 or 4)

(d)est: 00000000 — destination address (`dest`) used in the following commands

g(w) : 00000000 — gateway address (`gw`) used in following commands

p(u)t — adds new item (`dest` and `gw`), overwrites when an item with the same dest already exists

(g)et — for given dest finds the respective `gw`

(D)elete — deletes the item with given `dest` (if exists)

de(f)ault — writes `gw` into the `default gw` item (the `default gw` is used for un-listed destinations)

(c)lear — clears the whole table

(e)dit — edit mode - the whole table is transferred into Setr and edited or it is directly created in Setr. Then it is written into CU as one unit. See description bellow.

(r)ead — reads the whole table

type **r Enter**

```
ART No 1:
items: 2
default gw: C0A80052 (192.168.0.82   )
dest:    gw:
56509004 C0A86304 (86.80.144.4     192.168.99.4   )
56509005 C0A86412 (86.80.144.5     192.168.100.18 )
>>
```

Edit Art as one unit

From MORSE main menu type **Ae Enter Enter:**

```
Address Resolution Table editor:
ART No 1:
items: 2
default gw: 00000000 (0.0.0.0        )
dest:    gw:
56509004 C0A86304 (86.80.144.4     192.168.99.4   )
56509005 C0A86412 (86.80.144.5     192.168.100.18 )
(c)lear table (N)o:1
(d)est: 56509005 ga(t)eway: C0A86412
de(f)ault gateway
(g)et p(u)t (D)elete item
(r)ead (w)rite
(q)uit
```

The contents of the table is transferred into Setr. Using the commands `(c)lear, (d)est, ga(t)eway, de(f)ault, (g)et, p(u)t, (D)elete` modify the table in Setr similarly as was described above for work in menu `A Enter` in CU. Using `(w)rite` transfer the whole table into attached CU in Art designated by parameter `N(o):1`.

Similarly the content of table Art1 can be transferred to e.g. Art2 or to other CU. Example:

Read the content of Art3 from distant CU using command `(r)ead`:

```
Address Resolution Table editor:
ART No 3:
items: 1
default gw: 00000000 (0.0.0.0        )
dest:    gw:
11111111 33333333 (17.17.17.17     51.51.51.51    )
(c)lear table (N)o:3
(d)est: 0 ga(t)eway: 0
de(f)ault gateway
(g)et p(u)t (D)elete item
(r)ead (w)rite
(q)uit
690F8001h>
```

Don't close the Art menu, redirect to other CU, choice number of table `Art table (N)o:4` and write there the content by command (w)rite.

```
!h0101

690F0101h>N4

690F0101h>w

690F0101h>
ART No 4: O.K.
690F0101h>r

690F0101h>
ART No 4:
items: 1
default gw: 00000000 (0.0.0.0        )
dest:    gw:
11111111 33333333 (17.17.17.17     51.51.51.51    )
690F0101h>
```

It is visible by check `(r)ead` that the content is written in Art4.

13. Routing (retranslation) tables

Menu `r(T)ab` enables reading, editing and clearing of all routing tables in an CU.

From MORSE main menu type **T Enter**

```
Retranslation table:
  (l)ocal
  (w)ide area
  (n)et
  (g)lobal
(q)uit
>>
```

choose table type **w Enter**

```
Wide retranslation table No:
  (1) (2) (3) (4)
(q)uit
>>
```

choose table No: **1 Enter**

```
Retranslation table:
  (r)ead nontrivial paths
  (p)ath:0 via (n)ode:0
  (g)et p(u)t set (N)ode
  (c)lear  (e)dit
(q)uit
>>
```

(r)ead — reads the selected table; only the non-trivial items are displayed on the Setr.exe screen. Trivial items are those which are in the initial state after clearing the table, see the `(c)lear` command. In case there are too many non-trivial items in the table to be reported in a single packet, it is necessary to perform the `(r)ead` command more times to get the complete table.

(p)ath — the pertinent byte from the path destination address, corresponding to the selected type of table (global - net - wide - local) is entered

via (n)ode — part of `to` address from right side, length is for:

- local 1byte
- wide 2bytes
- net, global 4 bytes (complete address)

Both `(p)ath` a `(n)ode` are written in the table in Setr either by writing **(n)ode <value> Enter** or by **p(u)t Enter**. The `(n)ode` value is saved in the auxiliary buffer too. If the item is successfully written-in the service report is simply the ASCII string `"O.K."`

(g)et — displays the table item `(n)ode` for `(p)ath` entered, the `(n)ode` value is saved in the auxiliary buffer

p(u)t — puts the entered `(p)ath` and the `(n)ode` value from the auxiliary buffer in the table

(c)lear — clears all items in the selected table. In local tables all items are set to trivial paths, i.e. `01 to 01, 02 to 02,..., 3A to 3A,..., FF to FF`. In `wide, net` or `global` tables all items are set to zero values.

WALRUS - the Walrus firmware used in MRouter and MServer demands to perform (c)lear command before first use of the routing table.

set (N)ode — it is used for creating the time sync path with help of the macroinstruction:

- by command `T11` enter local routing table, read it by command `r`

```
Local retab. No 1
05 to:03  07 to:02  11 to:03
```

- put in the common part of address `>>N690f6511 Enter`

```
Retranslation table:
  (r)ead nontrivial paths
  (p)ath:7 via (n)ode:690F6511
  (g)et p(u)t set (N)ode
  (c)lear  (e)dit
(q)uit
>>
```

- the reference to the nearest node in direction to `690F6511` should be created for the `Uec` parameter:

```
>>p11 Enter

>>g Enter

>>
path  via  node
11  via  03
>> Enter

Retranslation table:
  (r)ead nontrivial paths
  (p)ath:11 via (n)ode:690F6503
  (g)et p(u)t set (N)ode
  (c)lear  (e)dit
(q)uit
>>
```

The address `690F6503` is created. The CU should the time sync questions send to this address. This address is located in the auxiliary buffer of Setr - clipboard.

- This address is set from clipboard to parameter `c` in `Ue` menu using command `cZ`.
- This procedure can be apply to all CUs in the network automatically with help of macro. The proper reference to the time source with regard to the routing tables is created in this way.

(e)dit — edit mode – other mode of table creating – The whole table contens is transferred to the Setr. The table is edited here and then is the table transferred to the distant CU again.

Example of edit without (e)dit command:

Every item path+node is transferred individually.

```
>>p03       Enter
>>n0300     Enter

O.K.              ...item p03 N0300 is written in CU

>>p05       Enter
>>n0501     Enter

O.K.              ...item p05 N0501 is written in CU
```

(g)et — displays the table item for `(p)ath` entered

```
>>p03       Enter
```

type **g Enter**

```
path  via  node
03  via  0300
>>
```

Editing of the complete table using (e)dit command:

For edit of whole table in the Setr press **e Enter Enter**. The `(r)ead` command runs automatically and shows the table content:

```
Retab editor
Wide retab. No 1
03to:0300 05to:0501 87to:8700
(c)lear table N(o):1
(p)ath 0 via (n)ode 0
p(u)t set (N)ode
(r)ead (w)rite
```

```
(q)uit
>>
```

The Art table in PC can be edited and then transferred to CU as one unit by command `(w)rite`.

03to:0300 — the table content in (distant) CU
05to:0501
87to:8700

(c)lear — clears the table in the Setr, not in the CU

table N(o):1 — the table number can be changed here. We can e.g. read the table 1, then change `N:3` and write it in CU as `Tl3`.

(p)ath 0 — the item path prepared for putting in the table in Setr

via (n)ode 0 — the item node prepared for putting in the table in Setr

p(u)t — `(p)ath` a `(n)ode` are written in the table in Setr either by writing **(n)ode <value> Enter** or by **p(u)t Enter**. Attention, the writing into CU must be done by `(w)rite` command!

red(i)rect — redirecting the table to other node, see example down

set (N)ode — see the menu above

(r)ead — reads the table from CU (and overwrite existing table in this menu)

(w)rite — writes the adjusted table to the CU

Rerouting using red(i)rect

Example - The connection to nodes 11, 12, 13 is directed via node 03 a connection to 15 via node 07:

```
Retab editor
Local retab. No 1
11 to:03  12 to:03  13 to:03  15 to:07
(c)lear table N(o):1
(p)ath 0 via (n)ode 0
p(u)t set  red(i)rect  (N)ode
(r)ead (w)rite
(q)uit
>>
```

The change in network occurred and the node 03 is now not accessible directly but via node 02, see the first item in next table. Now for as long as the items `(p):3, (n):2` are prepared, apply the `red(i)rect` command and correct all items 11, 12, 13 in the table from route 03 to route 02. The item `15 to: 07` stays unchanged.

```
Retab editor
Local retab. No 1
03 to:02  11 to:02  12 to:02  13 to:02  15 to:07
(c)lear table N(o):1
(p)ath 3 via (n)ode 2
```

```
p(u)t set  red(i)rect  (N)ode
(r)ead (w)rite
(q)uit
>>
```

The result table write in CU by command `(w)rite`.

14. Menu (c)nf

The Configuration menu serves for putting all the CU into default state and for the configuration backup. It is accessible from the Main menu using the command `c Enter`:

```
Configuration:
f(a)ctory setting
(b)ackup
r(e)store
(q)uit
>>
```

- Then by choosing **a Enter**, i.e. `f(a)ctory setting`, default values are inserted into all items (except the parameters and the calibration constants of the radio part). All items are inserted in the FLASH SRAM memory. The parameters are then copied into RAM after restarting the CU. The command `f(a)ctory setting` also leaves the content of the routing tables `r(T)ab` and content of the address resolution tables `(A)rt`, the cleaning of which is ordinarily undesirable.
- Commans `(b)ackup` and `r(e)store` are used in the MR400 series modems for the backup of SRAM memory.

15. Misc - supplementary functions

The menu `(m)isc` contains some auxiliary and service functions.

In the MORSE main menu type **m Enter**

```
Miscellaneous stuff:
(p)rotocol services
(t)ests
p(o)wer save po(w)er cycle
(b)att
```

(p)rotocol services — the supplementary functions for some protocols (Skao, Modbus-Skao, GPS, Modbus), they are used according to the description for respective protocol

(t)ests — the special tests used by some protocols, see the protocol description

p(o)wer save — switching of CU (MR400 only) into sleep mode with minimal energy consumption

(b)att — indicates the supply state. In MR25 modems it compares the supply voltage with the value `(U)cc limit` in the Communication unit menu. If the supply voltage is less than the `(U)cc limit` then the state is evaluated as the battery supply. The change is only indicated if it lasts for at least 60sec. Choose the function `(b)att` :

`Batt`

`>>Power supply: OL.` — OL indicates the network supplying. The report `Error No 1051` is generated when passing to this state and saved in Event log as `SUPPLY BACK EVT 1051`

`>>Power supply: OB.` – OB indicates the battery supplying. The report `Error No 1050` is generated when passing to this state and saved in Event log as `SUPPLY DROP EVT 1050`

16. Service

The Service part contains unit diagnostic tools, used mainly during the production process.

From MORSE main menu type **s Enter**

```
Service:
(h)eap info    (V)(v)ersion
RF service (R)X/(T)X    (e)cho
unit (s)tatus   (t)ime    (g)o
(c)lock info       loop (i)nfo
(r)adio mem c(o)mp ch(k) (m)obil
(q)uit
>>
```

16.1. (h)eap info

Up-to-date information about auxiliary buffers – address/size, useful for software debugging.

type **h Enter**

```
 Addr/size ...
 1C464h/  1602  1CB04h/  1602  1D1B4h/  1602  1D864h/  1602  1DEAEh/  2058
 1E6BEh/  2060  1EECEh/  2062  1F6DEh/  2064  1FEFEh/  2050  2070Eh/  2052
 20F1Eh/  2054  2172Eh/  2056  21F3Eh/  2058  2274Eh/  2060  22F5Ah/   144
 22FEAh/  1602
>>
```

16.2. (V) (v)ersion

If the capital letter V is chosen the following info is provided:

type **V Enter**

```
RACOM s.r.o.
Nove Mesto na Morave
Czech Republic

MR900 firmware : D 726
issued at : Aug 15 2005 21:00:18
serial    : 4857383
HW version: 3.00
>>
```

If the lower case v is chosen this info is given:

type **v Enter**

```
>>D 726
B 726
```

D 726 – firmware version of MORSE D (main software module E or D,G H..., 726 is the FW number)

B 726 – firmware version of MORSE B (base loader module)

16.3. RF service (R)X

Information from the RF receiver channel (mainly for production and debugging purposes)

type **R Enter**

```
Service RX:
RSSI constants: (a)djust de(f)ault
modem (s)tatus     RX e(y)e
fr(e)q. step  modem (R)eset
(q)uit
>>
```

(a)dj (a)djust de(f)ault — calibration setting for measuring the RSS level, for production purposes

(s)ta modem (s)tatus — informations about RF channel state

```
>>
RF modem status C0h
FX919 status    00h
idle errors 0  period 0 s
noise syncs 679  period 128 s
noise level -125 dBm
```

Rf Rf modem status C0h — status byte

FX FX919 status 00h — contents of the status register in the FX919 modem IC

idle idle errors 0 — The main processor administers supervision over the processor in the modem IC. If the modem do not carry out the instructions, a reset is performed and this counter is incremented

per period 0s — how often idle errors occur

sync noise sync 679 — a counter of modem false synchronisation events (to noise or interference on the RF channel)

per period 128s — how often noise sync occurs

nois noise level -125 dBm — the average noise level on the RF channel. The first 3 ms of each received packet and the alien packets weaker then RSS treshold are included in the noise. The value is integrated in the time, the influence of disturbing signal expires after about 10 sec.

e(y)e RX e(y)e — This function is used for diagnostic purposes in the modem part (in production only).

fr(e) fr(e)q. step — Function used to switch channels in the set grid (25kHz). Both receiver and transmitter frequencies are changed using the `(u)p` or `(d)own` command.

```
Freq. (s)tep: 25 (u)p (d)own
(q)uit
```

(R)es modem (R)eset — RF modem reset performed. Any service function on the RF channel is completed by performing a reset of the RF modem.

16.4. RFservice (T)X

Services of the RF transmitter channel for production purposes.

type **T Enter**

```
Service TX:
(c)arrier (r)andom (s)ync
(dNNNNNNNN) hex data (p):5000ms
fr(e)q. step modem (R)eset
(q)uit
```

(c)ar (c)arrier — Radio part transmits unmodulated carrier.

(r)an (r)andom — Continuous transmission of random data. Used in production to judge the signal with the RX eye at the receiving station.

(s)yn (s)ync — Transmits the RF data frame synchronisation sequence.

(dNN) (dNNNNNNNN) — Repeated transmission of four bytes. From these, each is assigned another pair of NN. This enables measurements with an arbitrary setting.

(p): 5 hex data (p): 5000ms — transmission duration, max. selectable time is 65 535 ms

fr(e) fr(e)q. step — see RF service (R)X above

(R)es modem (R)eset — see RF service (R)X above

16.5. (e)cho

Toggles the keyboard echo off/on.

type **e Enter**

(e)cho ECHO OFF now — the keyboard characters are not displayed now

To switch ECHO ON again, type **e Enter** here or from anywhere press **Esc Enter se Enter** or restart `Setr.exe`

(e)cho ECHO ON now — the keyboard characters are displayed

16.6. unit (s)tatus

type **s Enter**

```
local time:2005-09-01 15:12:45 LOC/daylight savings; uptime:537 sec
  day time:2005-09-01 13:12:45 GMT; log write at:86400
B0 30001014;
CNFDB use:81.3%;size:4076;items:71;banks:1;

All transmitting disabled!
Test active on Node 1
Binmon: SCC2

HEAP: 12/640 objs, 1034632 bytes free
DBUFS: 18/64 objs, 35840 bytes free
STACK: 2565 bytes free?; Cold Start

    |SCC    |RFC      |ETH
    |0 1 2 3|0 1 2 3 4|0
ret  0 0 0 0 0 0 0 0 0 0 - -
user 0 0 0 0 0 0 0 0 0 0 - -
Qbuf 1 0 0 0
TX   0 0 0 0           0
RX   1 0 0 0           0
flTX 1 1 1 1
flRX 1 1 1 1
RTS  0 0 0 0
DTR  0 0 0 0
err buf 0; Service mode.
>>
```

local	local time: 2005-09-01 15:12:45 LOC — value according to menu `(s)ervice (t)ime (d)ay time` with suffix `LOC` for local time (winter) or `LTS` for local summer time
uptime	uptime: 537 sec — time elapsed from hw reset, see sts menu
day	day time:2005-09-01 13:12:45 GMT — GMT time maintained in Communication Unit, see `(U)nit (e)dit`
log	log write at 86400s — time counted from hw reset in which the next log write will occur, see `(U)nit (e)dit (l)og period`
	The unit status would also display warnings when the CU is in some exceptional state, e.g.:
All	All transmitting disabled! — non standard states messages
Test	Test active on Node 1 — message about running some of tests on the node 1
Binmon: SCC2	Binmon: SCC2 — monitoring of serial channel
HEAP	remaining free space for auxiliary buffers. The table displayed shows the number of packets (buffers) waiting in different queues throughout the CU.

16.7. (t)ime

There are three sort of time used:

- `The GMT` is used in the CU and is kept with a lithium battery. Effective life of the battery is minimally 15 years under normal operation. The life at disconnected supply is however 3 years only.
- `The local time` valid on the CU station. This time is set by `(s)ervice (t)ime`. It is corrected by `(U)nit (e)dit DI(F)F` parameter to get the GMT, which is kept in the CU.
- `The distant operator time` jis used by the operator, which can be situated in different time zone. It is obtained from GMT by subtracting of parameter `-t`, kwhich is found in the file `Morse.par` in the operator's PC.

The local time us used in `(s)ervice (t)ime` menu and in transmitted old monitoring, which creates the monitoring message completely in the source node. The GMT is used in all other functions inside the CU, like saving time of logs, Events or time synchronization by `(U)nit (e)dit Time sync`.

The distant operator time is used for display of logs, Events and (new) Monitoring messages.

type **t Enter**

```
Time:
(d)ay time   (s)econds
set (n)ew time
dat(e)  set ne(w) date
(q)uit
>>
```

(d)ay (d)ay time — 24h time information `(hh:mm:ss.ms)`

`21:17:25.85`

(s)ec (s)econds — the number of seconds which have run since the last hardware reset

`9731`

(n)ew set (n)ew time — to insert time 9h 15m 00s type:

n09:15:00 Enter, inserting is acknowledged by the message:

`New time accepted`

dat(e) dat(e) — date information

`07.09.2005`

ne(w) set ne(w) date — to insert date 1. 10. 2005 type:

w01.10.2005 Enter

`New time accepted`

16.8. (g)o

Starts one of the CU software modules.

type **g Enter**

go MORSE (E) — restarts the main module (warm start)

go MORSE (B) — starts the base loader module the same way as during a cold start (equal to switching on the power supply)

go MORSE (A) — starts the air loader module (if present) used for remote loading of the main module via the RF channel

go MORSE (W) — starts the wire loader module (if present) used for remote loading of the main module via the SCC channel

power supply (S)hutdown — switch off the power supply for 15 seconds, then switch on again - obsolete, not used

16.9. (c)lock info

The system clock frequency is measured in the course of four seconds with resolution in kHz. Caution! The CU is "dead" for these four seconds. In real operation, this action will cause an CU failure for four seconds. The signal from the on-board RTC (Real Time Clock) is used for measurement, so the accuracy of the clock value obtained cannot be better than the RTC accuracy.

type **c Enter**

```
System clocks -
configured value:         66666667 Hz
measured by on board RTC: 66662400 Hz
>>
```

16.10. loop (i)nfo

The time duration for running the program main loop.

type **i Enter**

```
main loop time statistics
for last 23 seconds:
longest run 3 ms
average run 0.053 ms
>>
```

16.11. (r)adio mem c(o)mp ch(k)

Parameters for production purposes.

16.12. (m)obil

The current state of mobile connection.

type **m Enter**

```
Mobile stuff
Center:

(a)ddr:00000000h (m)ask:00000000h
(c)enter status
Mobiles at RF:
 (0) (1) (2) (3) (4)
(q)uit
>>
```

(c)en (c)enter status — in Mobile Center available only, offers the list of the active Mobile stations and the Base stations currently belonging to each Mobile station

```
>>
Nr   Addr      Base      Timeout  CnfTim
  1  690F1001  690FC000    0294    0304
== send:1  stored:1  mobiles ==
>>
```

CnfTim — maximal time of record validity according to `Ne1v` parameter in the Center CU

Timeout — actual time of record validity

stored: — number of stored records (number of active Mobiles)

send: — number of sent records is limited, if number of stored Mobiles is bigger then number of sent, then use the filter

`(a)ddr:00000000h (m)ask:00000000h`
which brings the chosen Mobile addresses only

(1) Mobiles at RF: (0) (1) (2) (3) (4) — in Mobile CU only, choose the number of RFC running in Mobile mode

```
Mob
>>
RF protocol HCSMA (Hruska's Carrier Sense Multiple Access)
mobile info on RF channel 1:
RSSI -70 dBm
base 690FC000
>>
```

base 690FC000 — the current Base station which mediates the connection to Center

RSSI -70 dBm — received signal strength from Base CU

17. Diagnostic

All diagnostic services of the CU can be accessed from this menu. A short description of the main groups of diagnostic tools follows in the respective paragraphs. From MORSE main menu type **`i Enter`**

```
Diagnostic:
  (M)onitoring/old (m)on
  (s)ystem channels
  stat (l)ogs (c)ron
  (E)vent/e(r)ror log
  (t)ests (R)ecorder
  channel se(n)d
  enable RF(T)X
(q)uit
>>
```

(M)on (M)onitoring and (s)ystem channels — these menus complement each other. Monitoring allows trace the communication channels. It is configured individually for the radio channels RFC, serial channels SCC, ethernet channel ETH and for the network agent channels NAG. The packets can be followed on the external (physical) interface or on the internal (CNI). The menu (s)ystem channel defines, where are the monitoring messages sent.

(m)on old (m)on — old version of monitoring used till firmware version 550

(l)og stat (l)ogs — statistical logs record the number of packets passed through individual channels including the diagnostic informations

(E)ve (E)vent — recording of the events in CU like writing in the memory, reading, switch the modem off and on and others

(t)es (t)ests — tests used for the network diagnostics

se(n) channel se(n)d — packet simulation on the serial interface

RF(T) RF(T)X — disable and enable of the radio channel transmissions

18. Monitoring

Valid for software version 551 and higher.

Monitoring enables data flow through various interfaces in the CU to be followed. Monitoring information is arranged in the source Node into a binary file and dispatched. In Setr or another application on the Destination address the binary data is compiled into the required form. In this case they attain the form of the monitoring reports described below.

By means of the Monitoring menu it is possible to choose the interface followed, the destination Node the binary data packets are sent to and the form in which these packets are displayed on the Setr.exe display.

From MORSE main menu type **i M Enter**:

```
Monitoring:
(S)CC R(F)C (E)TH NA(G) o(b)solete
(o)ff
(f)eatures
(q)uit
>>
```

The direction and parameters of transmission of monitoring packets are chosen in the separate menu `d(i)ag (s)ystem channels`, on/off and filtration of individual channels are contained in the next menus. Initialisation is performed in each menu separately.

(S)CC — filter for serial channels

R(F)C — filter for radiofrequency channels

(E)TH — filter for Ethernet channel

NA(G) — filter for Network Agent channel

o(b)s - o(b)solete — obsolete filter for Node input and output is replaced by new features of other items

(o)ff — Sets off the monitoring in the way that all items (R)X and (T)X in RAM and in flash memories are switched off. Other parameters remain unchanged. Running monitoring can be stopped at any moment with the commands:

- `Q Enter`
- `iMo Enter`

(f)eat - (f)eatures — other features setting

Content menu (f)eatures:

```
Monitoring:  (d)isplay:HEX
SCC e(v)ents:ASCII
RFC (c)ompress:UNCOMPRESS
(T)ime format:ABSOLUTE
```

```
RSS (B)eep:OFF  (L)ow freq. (-120dBm):400Hz  (H)igh freq. (-40dBm):4000Hz
RSS (S)hort:OFF
(q)uit
>>
```

(d)isp - (d)isplay — switches the way of displaying messages from monitored channels. The commands `(d)isplay` and `SCC e(v)ents` are set into RAM by merely writing in Setr. These parameters are not saved into flash memory. The parameters only influence the CU receiving the monitoring.

`(h)` - HEX — displays data in the hexadecimal form

`(a)` - ASCII — displays data in the ASCII code

`(r)` - ROW — hexadecimal form without gaps

e(v)en - SCC e(v)ents — the current parameters of async.link (RTS, CTS...) are displayed according to the choice:

`(a) ASCII` – displays in ASCII only when the parameter is changing

`(h) HEX` – displays in hexadecimal form in each monitoring message

`(A) ASCII ALL` – displays in ASCII in each monitoring message

(c)om - RFC (c)ompress — the data in RF channel:

`(u)` - UNCOMPRESS — normal (long) form of data

`(n)` - DO NOT MODIFY — compressed form which is transferred via the radio channel

(T)ime - (T)ime format — the time of monitored message can have two forms:

`(a)` - ABSOLUTE — local time in CU

`(r)` - RELATIVE — the increment from previous message in ms

(B)eep - RSS (B)eep — the RSS level of monitored packet can be indicated by sound of various pitch

`(L)ow` - (L)ow freq.(-120dBm): 400Hz — the optional sound frequency for week signal

`(H)igh` - (H)igh freq.(-40dBm): 4000Hz — the optional sound frequency for strong signal

(S)hort - RSS (S)hort — the short form of message from physical RF channel is available

18.1. Routing of the monitoring messages

In this separate menu the source, destination and in what way monitoring messages will be transmitted are determined. The setting is common for all channels. The menu enables preparation of two variations of `id` which can be independently assigned to the individual monitoring channels. The choices `id0` and `id1` are suitable for monitoring, because they allows switch it off by the common command `iMo`.

From MORSE main menu type **ise Enter Enter**:

```
System channels:
(Service 'iMo' works for s0 and s1 only)

id|--Node--addr------timeout---size---s(e)c--
(0)   0    00000000     888     400     ON
(1)   0    00000000     888     400     ON
(2)   0    00000000     888     400     ON
(3)   0    00000000     888     400     ON
(4)   0    00000000     888     400     ON
(5)   0    00000000     888     400     ON

de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
>>
```

Also choose **0 Enter**

```
System channels:
(N)ode:0 Destination (a)ddr.:00000000
Destination (2)addr.:00000000 (3)addr.:00000000 (4)addr.:00000000
            for splitting packets only
(t)imeout:888ms
max. TX (s)ize:400byte
s(e)curity:ON
(I)nit (W)rite
(q)uit
>>
```

(N)od - (N)ode:0 — The choice of the Node in the traced CU which generates the monitoring packets. The monitoring reports are then labelled by its address. It is suitable here to choose Node 1. This Node should have the connection (direct or by the routing tables) to the `Destination (a)ddr`.

(a)dd - Destination (a)ddr: 00000000 — The monitoring packets are sent to this address (in local or distant CU). The monitoring packets leave the Node which has this address via the service output. The channel to which the application with monitoring is connected must be assigned to this output.

Typically Setr is connected via the service cable, i.e. in the menu `(N)odes` column `(s)` the channel SCC2 to which the service cable is connected is set.

If the Split is chosen, e.g. in the menu `iMEIe 0uSn` , the copy of the packet running is set to this address.

(2)ad - Destination (2)addr.: 00000000 — Next 3 addresses, where the packet copy from Split function can be sent.

(t)im - (t)imeout:888ms —The longest time over which data is collected in the monitoring packet, before it is sent to the `Destination (a)ddr`.

(s)ize - (s)ize:400byte — The maximum length of the sent monitoring packet. After it is filled up the packet is dispatched and the next packet starts to be made up.

The above-mentioned data can be prepared in two versions. Selection of one of them should be made in the respective monitoring channels.

After parameter setting carry out `(W)rite, (I)nit`. The command `iMo` does not have an effect on the contents of the `(s)ystem channels` menu.

18.2. Monitoring SCC channels

The monitoring parameters of SCC channels are set here. Setting is performed individually for each channel.

V MORSE main menu type **iMS Enter**:

```
SCC monitoring:
(p)hysical layer  CN(I)
(q)uit
>>
```

This menu, like the menus for other monitored channels, offers two sets of monitored points:

`(p)hy` - (p)hysical layer — monitoring of packets on the SCC physical layer, i.e. packets coming to and from the connected equipment. This kind of monitoring was used in sw449 and older versions.

`CN(I)` — monitoring of packets exchanged between the SCC and the Node.

18.2.1. SCC monitoring – physical layer

After choosing **iMSpe Enter Enter** we obtain the menu for the physical layer:

```
  SCC monitoring:

SCC--s---RX--TX--Ev----------len--------
(0)  0   ON  ON  OFF         32
(1)  0   OFF OFF OFF         32
(2)  0   OFF OFF OFF         32

de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
>>
```

For choosing the individual Serial Communication Channel type e.g. **0 Enter**:

```
SCC monitoring:
O(N) O(F)F
o(n) o(f)f          (R)X:ON  (T)X:ON  (E)vents:OFF
(s)ys. channel:0
```

```
(l)ength:32
(I)nit (W)rite
(q)uit
>>
```

O(N) - O(N) O(F)F — the possibility of simultaneously switching on RX and TX in the chosen channel.

(R)X - (R)X:ON — Switching on / off the monitoring in the Receiving direction, i.e. packet arrival through the SCC channel from outside to the CU.

(T)X - (T)X:OFF — Switching on / off the monitoring in the Transmitting direction.

(E)v - (E)vents:OFF — switching on/off the monitoring of the hw signals on the port, e.g. CTS, RTS

(s)ys - (s)ys.channel:0 — Choose method of sending according to `(s)ystem channels` menu.

l(e)n - data l(e)ngth:32 — Max. length of data field in the traced packet, which is transmitted and displayed behind the monitoring report. A value from 0 to 1500 bytes can be chosen.

The format of a monitoring report for SCC channels – physical part:

```
Monitoring: source 690F8700|6.
O.K.
>>
07:27:35.532 rx    4 | S00
AAAA AAAA

Monitoring: source 690F8100|7.
07:27:36.679 tx   18 | S01
400E 00A9 690F 8909 690F 8700 AAAA AAAA 4EAE
07:27:36.701 rx    2 | S01
8106
```

The meaning of individual parts of a report:

...source... - Monitoring: source 690F8700 1. — Labelling of a Node which is the source of the report and the report order number from this Node. This src address is only displayed in the case when it was changed in monitoring message.

07:27:35.532 — time stamp (hh:mm:ss.msec)

rx — the direction of packet progression, here from external source through SCC to CU

4 — dec – length of monitored packet

S00 — monitored serial channel

AAAA AAAA — transmitted packet, here simple data from async. link, in the second case the data wrapped up in the retranslation Morse packet, in third case it is ACK

SCC monitoring – CNI layer

This menu is accessible from MORSE main menu by command **iMSIe Enter Enter**:

```
    CNI Monitoring:
          |         retranslation         |            user              |
    s-moR |-RX--TX--ERR-len-type-mask-Spl-|-RX--TX--ERR-len-type-mask-Spl|
(0) 0 OFF | OFF OFF OFF   0 0000 0000 OFF | OFF OFF OFF   0 0000 0000 OFF|
(1) 0 OFF | OFF OFF OFF   0 0000 0000 OFF | OFF OFF OFF   0 0000 0000 OFF|
(2) 0 OFF | OFF OFF OFF   0 0000 0000 OFF | OFF OFF OFF   0 0000 0000 OFF|

de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
>>
```

The part `retranslation` is used for monitoring traffic between the retranslation SCC input/output and the Node. The part `user` is used for packet tracking between the SCC user input/output and the Node.

- The columns `s,moR` are common for both parts.
- The direction of packet movement is expressed from the SC channel point of view, i.e. the packet going from SCC to the Node is called TX here.

By selecting the SCC number we enter the next menu containing the address filters:

type **0 Enter**

```
SCC CNI Monitoring:
(s)ys. channel:0
(r)etranslation
(u)ser
(d)st :00000000  (D)st mask :00000000 (1)-for :RX
sr(c) :690FABCD  sr(C) mask :FFFFFF00 (2)-for :TX
t(o)  :00000000  t(O) mask  :00000000 (3)-for :RX
fro(m):00000000  fro(M) mask:00000000 (4)-for :RX
use MO(R)SE addresses:OFF
(I)nit (W)rite
(q)uit
>>
```

This menu is fully applicable for the retranslation part. Only the `(d)st` and `sr(c)` addresses are useful for the user part.

(s)ys - (s)ys. channel:0 — Choose method of sending according to `(s)ystem channels`

(d)st ... - (d)st: (D)st mask: — Only the packets that fulfil all set conditions are displayed. For example the transmitted packets must have the source address in the range from `690FAB00` to `690FABFF` to be displayed. The received packets will be displayed without exception.

MO(R) - use MO(R)SE addresses:OFF — addresses type used for displaying and filtration

OFF — the user addresses are used for the display and filtration purposes

ON — the MORSE addresses are used, see the address translation in the menu SIe

For the retranslation part of menu type **r Enter**:

```
Ret. CNI monitor:
O(N) O(F)F          (R)X:OFF  (T)X:OFF  TX (E)rr:OFF
eliminate (d)eadlock:ON  (S)plit:OFF
(l)ength:0
Packet (t)ype:0000  type (m)ask:0000
(q)uit
>>
```

O(N) - O(N) O(F)F — The possibility of simultaneously switching on RX and TX in the chosen channel.

(R)X - (R)X:ON — Switching on / off the monitoring in the Receiving direction, i.e. arrival of packet from the Node to the SCC.

(T)X - (T)X:OFF — Switching on / off the monitoring in the Transmitting direction.

(E)rr - TX (E)rr:OFF — Switching on / off the monitoring of the error messages.

(d)ead - eliminate (d)eadlock: ON — Skip monitoring of the monitoring messages to prevent channel deadlock.

(S)plit - (S)plit:OFF — The Split ON function changes monitoring to splitting. The copy of packet is created and sent to address `ise2a` or to addresses `ise22`, `ise23`, `ise24` also. The `id2` to `id5` from `ise` menu is used for this purposes.

(l)en - (l)ength:0 — Max. length of data transferred.

(t)ype - Packet (t)ype: 0000 type (m)ask: 0000 — The desired packet type can be determined here.

The user part of menu:

type **u Enter**

```
User CNI monitor:
O(N) O(F)F          (R)X:ON  (T)X:ON  TX (E)rr:OFF
eliminate (d)eadlock:ON  (S)plit:OFF
(l)ength:40
Packet (t)ype:0000  type (m)ask:0000
(q)uit
>>
```

The meaning of items is similar as in the previous menu. The only difference is that this menu is valid for communication between the SCC user input/output and the Node.

The format of a monitoring report for SCC channels – CNI layer:

```
15:18:31.791|690F120E 690F120E|690F120E 690F1241|S00I  OUT   2|09 1dat
AAAA
15:18:31.792|                |690F120E 690F1241|S01I   IN   2|09 1serv
AAAA
```

Packet `AAAA` in this example arrived via the async.link to the SCC0 channel and was monitored by retranslation monitoring on its way from SCC0 to the node `690F120E` (OUT direction). The packet is monitored on the CNI level and this is why it contains only the data without the packet head. Then this packet was sent from the node to the user channel SCC1 (IN) and was monitored at its input into SCC1 by user monitoring.

Meaning of items:

15:18:31.791 — time stamp

|690F120E 690F120E |690F120E 690F1241 — addresses in the order - to, from, destin, source

|S00I OUT 2 — channel monitored, direction, length of the packet

|09 1dat — packet type, serial number 0 to 7, kind of packet

18.3. Monitoring RFC channels

From MORSE main menu type **iMF Enter**:

```
RFC monitoring:
(p)hysical layer  CN(I)
(q)uit
>>
```

This menu, like the menus for other monitored channels, offers two sets of monitored points:

`(p)hy` - (p)hysical layer — monitoring of packets on the RFC physical layer, i.e. packets coming to and from the radiomodem.

`CN(I)` — monitoring of packets exchanged between the RFC and the Node.

18.3.1. RFC monitoring – physical layer

After choosing **iMFpe Enter Enter** we obtain the menu for the physical layer:

```
RFC monitoring:
o(N) o(F)f          (R)X :ON   (T)X :OFF  Medi(u)m:OFF
(s)ys. channel:0
(l)ength:100     rx (p)romisc. lvl:0 (normal operation)
Filter:
Only (h)ead crc OK :OFF
Only d(a)ta crc OK :OFF
Packet t(y)pe:0008 tmas(k):0008
```

```
(d)st :690F8700  (D)st mask :00000000 (1)-for :RX
sr(c) :00000000  sr(C) mask :00000000 (2)-for :TX
t(o)  :690F8700  t(O) mask  :FFFFFFFF (3)-for :RX
fro(m):690F8100  fro(M) mask:FFFFFF00 (4)-for :RX
de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
>>
```

o(N) - o(N) o(F)f — The possibility of simultaneously switching on RX and TX in the radio channel.

(R)X - (R)X :OFF — Monitoring on / off in the `Receiving` direction.

(T)X - (T)X :OFF — Monitoring on / off in the `Transmitting` direction.

Medi(u)m — The RSS value and the RF modem status is sampled every 8 milliseconds and coded into a 4-bit number (see RF medium log description for details). When RF medium monitoring is switched on, a monitoring message is generated every second, containing these 4-bit samples from the previous second.

(s)ystem - (s)ystem channels: 0 — The choice of report transmitting according to `(s)ystem channels` menu.

(l)en - (l)ength: 100 — Max. length of data in the packet transmitted.

(p)rom - rx (p)romisc. lvl:0 (normal operation) — the choice of category of packets which are monitored:

(0) – 0 (normal operation) — normal traffic

(1) – 1 (includes foreign headers) — normal traffic include alien packets

(2) – 2 (includes crc errors) — normal traffic include packets with wrong CRC of head

(3) – 3 (sync to foreign packets + data) — all radio signals

For restricting the amount of data transmitted the filters are used which distinguish:

- exactness of CRC check sum
- packet type
- addresses

The displayed packet must fulfil the conditions of all 7 filters, if they are switched on:

- check sum

  (h)ead - Only (h)ead crc OK :OFF — CRC for packet head is OK (the filter is off here)

  d(a)ta - Only d(a)ta crc OK :OFF — CRC for packet data is OK (the filter is off here)

- packet type

t(y)pe - Packet t(y)pe: 0008 — packet type

mas(k) - tmas(k):0008 — the mask for packet type application (`0000 = off`). The mask is indicated by ones, which packet type bits in the monitored packed must be equal to the parameter `Packet t(y)pe`.

The example used here:
`Packet t(y)pe:0008 tmas(k):0008`
can be useful for choosing the packets containing user data (type 0x09, 0x89) and the echo packets generated after an exclamation sign command (0x98, 0x9A). The ACK packets having type 0x06 are not monitored.

- 4 filters for addresses

```
(d)st :690F8700  (D)st mask :00000000 (1)-for :RX
sr(c) :00000000  sr(C) mask :00000000 (2)-for :TX
t(o)  :690F8700  t(O) mask  :FFFFFFFF (3)-for :RX
fro(m):690F8100  fro(M) mask:FFFFFF00 (4)-for :RX
```

The packets are classified in accordance with addresses `destination, source, to, from` and with the direction of their progression towards a monitored CU (received RX, transmitted TX). For a received packet conditions are valid which have `RX` at the end of the line. Conditions containing TX are ineffective for it. For the transmitted packet the opposite is valid. In this example the received packet, to be monitored, must contain the address (`to`) `690F8700`, the (`from`) address must be in the range `690F8100` to `690F81FF`, the (`source`) address is not valid for it, the (`destination`) address can be arbitrary.

(d)st: 690F8700 — Required destination address

(D)st mask: 00000000 — Mask (allows arbitrary address)

(1)-for :RX — Valid for received packets

sr(c): 00000000 — Required source address

sr(C) mask: 00000000 — Mask (allows arbitrary address)

(2)-for: TX — valid for transmitted packets

t(o): 690F8700 — Required to address

t(O) mask :FFFFFFFF — Mask (requires precise congruity)

(3)-for: RX — valid for received packets

fro(m): 690F8100 — Required address from

fro(M) mask: FFFFFF00 — Mask (requires precise congruity in the parts global, net, wide)

(4)-for: RX — valid for received packets

The format of a monitoring report for RFC – physical layer:

```
Monitoring: source 565098C6|5.
RF mon      |toa      frm     |dst      src     |lNo!DQ!RSS size|TT N
07:14:47.585|565098C6 565098BA|565098C6 565098BA|015*31* 77   10*98 4dat
E018 C100 5650 98BA 4201
07:14:47.585|565098BA 565098C6|                 |015  RFTX     0 06  ack
07:14:47.742|565098BA 565098C6|565098BA 565098C6|00E  RFTX    26 9A 4dat
E018 C100 5650 98BA 4201 7F4D 5650 98C6 8000 8000 5650 98C6 4201
07:14:47.855|565098C6 565098BA|                 |00E*31~ 77    0*06  ack
```

The meaning of individual parts:

...source... - Monitoring: source 565098C6|5 — Node, which is the source of the report and the report order number from this Node. This is displayed only when the source address changes.

RF mon.... — the header appears in first monitoring message after start Setr only

07:14:47.585 — timestamp

565098C6 — to adress

565098BA — from adress

565098C6 — destination adress

565098BA — source adress

|015 — order number

31 77 10* — receiving from outside to the RF channel, DQ, RSS:

- * header CRC is OK or
- ! header CRC wrong
- 31 DQ
- * data CRC OK or
- ! data CRC wrong or
- ~ no data, e.g. for ACK
- 77 RSS
- 10 dec – length of monitored packet in bytes
- * Rx, packet is received

or alternatively:

RFTX 26 — transmission from the radio to the outer space

RFTX Tx, packet is transmitted

	26 dec – length of monitored packet in bytes
98	— packet type
4	— order number
dat	— the kind of packet
E018...	- E018 C100 5650 98BA 4201 — the monitored packet

RFC monitoring – CNI layer

This menu is accessible from the MORSE main menu using the command **iMFIe Enter Enter**:

```
    CNI Monitoring:
          |         retranslation         |            user              |
    s-moR |-RX--TX--ERR-len-type-mask-Spl-|-RX--TX--ERR-len-type-mask-Spl|
(0) 0 OFF | OFF OFF OFF   0 0000 0000 OFF | OFF OFF OFF   0 0000 0000 OFF|
(1) 0 OFF | ON  ON  ON   30 0000 0000 OFF | OFF OFF OFF   0 0000 0000 OFF|
(2) 0 OFF | OFF OFF OFF   0 0000 0000 OFF | OFF OFF OFF   0 0000 0000 OFF|
(3) 0 OFF | OFF OFF OFF   0 0000 0000 OFF | OFF OFF OFF   0 0000 0000 OFF|
(4) 0 OFF | OFF OFF OFF   0 0000 0000 OFF | OFF OFF OFF   0 0000 0000 OFF|

de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
>>
```

The behavior **iMFIe** menu is analogical like in **iMSIe** menu in the section SCC monitoring - CNI layer. The part `retranslation` is used for monitoring traffic between the retranslation RFC input/output and the node. The part `user` is not used for the RF channel.

The direction of packet movement is expressed from the RF channel point of view, i.e. the packet going from the RFC to the node is called TX here.

By selecting the RFC number we enter the next menu containing the address filters:

1 Enter

```
RFC CNI Monitoring:
(s)ys. channel:0
(r)etranslation
(u)ser
(d)st :00000000  (D)st mask :00000000 (1)-for :RX
sr(c) :00000000  sr(C) mask :00000000 (2)-for :RX
t(o)  :00000000  t(O) mask  :00000000 (3)-for :RX
fro(m):00000000  fro(M) mask:00000000 (4)-for :RX
use MO(R)SE addresses:OFF
(I)nit (W)rite
(q)uit
>>
```

(s)ys - (s)ys.channel:0 — Choose method of sending according to the `(s)ystem channels` menu.

(r)et - (r)etranslation — This menu is applicable for the retranslation part.

(u)ser — This menu is not used for RF channel.

(d)st - (d)st and next filters — Only the packets which fulfil all set conditions are displayed. See the previous paragraph `iMF(p)hysical layer`.

MO(R) - use MO(R)SE addresses — `OFF` - not used, in `FIe` the address translation is not used

The retranslation part of the menu:

type **r Enter**

```
Ret. CNI monitor:
O(N) O(F)F          (R)X:ON  (T)X:ON  TX (E)rr:ON
eliminate (d)eadlock:ON  (S)plit:OFF
(l)ength:30
Packet (t)ype:0000  type (m)ask:0000
(q)uit
>>
```

O(N) - O(N) O(F)F — The possibility of simultaneously switching on RX and TX in the chosen channel.

(R)X - (R)X:ON — Switching on / off the monitoring in the Receiving direction, i.e. packet arrival from the Node to the RFC.

(T)X - (T)X:ON — Switching on / off the monitoring in the Transmitting direction.

(E)rr - TX (E)rr:ON — Switching on / off the monitoring of the error messages.

(d)ead - (d)eadlock:ON — Skip monitoring of the monitoring messages to prevent channel deadlock.

(S)plit - (S)plit:OFF — The function Split ON changes the monitoring to the splitting, see the SCC monitoring.

(l)en - (l)ength:30 — Max. length of data field in the traced packet.

(t)ype - (t)ype: (m)ask: — The desired packet type can be determined here.

The format of a monitoring report for RFC – CNI layer

```
Monitoring: source 565098C6|3.
CNI mon     |toa      frm     |dst      src     |          size|TT N
09:53:32.450|565098C6 565098BA|565098C6 565098BA|R01I   OUT   10n98 7dat
09:53:32.450|565098BA 565098C6|565098BA 565098C6|R01I    IN   26N9A 7dat
```

This example was scanned without data, `(l)ength:0`. The first line contains the received packet coming from RFC1 to the Node. The second one displays the same packet after processing in the

Node, which is sent to RFC1 to be retranslated again. Note, that there are no ACK packets here because they are generated by RFC and they are not sent to the Node.

Meaning of items:

09:53:32.450	— time stamp
|565098C6 565098BA|565098C6 565098BA	— addresses in the order - to, from, destin, source
|R01I OUT 10	— channel, direction, and length of the packet monitored
n98 7dat	— packet type, serial number 0 to 7, kind of packet

18.4. Monitoring ETH channels

From MORSE main menu type **iME Enter Enter**:

```
ETH monitoring:
(p)hysical layer  CN(I)
(q)uit
>>
```

This menu like the menus for other monitored channels offers two sets of monitored points:

`(p)hy` - (p)hysical layer — Monitoring of packets on the ETH physical layer, i.e. packets coming to and from the Ethernet line.

`CN(I)` — Monitoring of packets exchanged between the ETH channel and the Node.

18.4.1. ETH monitoring – physical layer

After choosing **pe Enter** we obtain the menu for the physical layer:

```
ETH monitoring:
O(N) O(F)F          (R)X :ON (T)X :ON
(s)ys.channel:0
(l)ength:100
(P)romisc. mode:OFF
IPFilter:
ip (d)st :CA000001  ip (D)dst mask :FFFFFFFF (1)-for :RX
ip sr(c) :00000000  ip sr(C) mask :00000000 (2)-for :RX
de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
>>
```

O(N) - O(N) O(F)F — the possibility of simultaneously switching on RX and TX in the ETH channel.

(R)X - (R)X: ON — Switching monitoring on/off in Receiving direction.

(T)X - (T)X: ON — Switching monitoring on/off in Transmitting direction.

(s)ys - (s)ys.channel: 0 — The choice of report transmitting according to `(s)ystem channels` menu.

(l)en - (l)ength: 100 — Max. data length in traced packet. It is possible to choose from 44 to 1500 bytes here. The IP and UDP heads and flags are always displayed.

(P)ro - (P)romisc. mode — choice of monitored packets

OFF — displays the packets for own Eth channel only

ON — displays all packets

(d)st - ip (d)st: C0A00001 — **The address filter**. Destination IP address which must be identical to the packet's destination address in the range of the mask.

(D)st - ip (D)st mask: FFFFFFFF — The mask for the destination address. If the mask contains some zeros then the address `ip (d)st` must also contain zeros in the corresponding positions.

(1) - (1)-for: RX — The condition is valid for received packets.

sr(c) - ip sr(c): 00000000 — Source IP address which must be identical to the packet's source address in the range of the mask.

sr(C) - ip sr(C) mask: 00000000 — The mask for the source address, here the condition is not determined.

(2) - (2)-for: RX — The condition is valid for received packets.

The format of an ETH monitoring report – physical layer

```
12:58:51.826|eth:RX   60 |FFFFFFFFFFFF |0002A94C9D8F | ARP/REQ
0001 0800 0604 0001 0002 A94C 9D8F C0A8 0001 0000 0000 0000 C0A8 0009 C0A8
12:58:51.827|eth:RX   60 |0002A94C9D8F |0002A94CA177 | ARP/ANS
0001 0800 0604 0002 0002 A94C A177 C0A8 0009 0002 A94C 9D8F C0A8 0001 C0A8
12:58:51.829|eth:TX   68 |0002A94CA177 |0002A94C9D8F | IP/UDP/MOR/RET/DAT
4500 0036 009B 4000 4011 B8C1 C0A8 0001 C0A8 0009 22B8 22B8 0022 3B63 D200
12:58:51.834|eth:RX   60 |0002A94C9D8F |0002A94CA177 | IP/UDP/MOR/RET/CTL
                                                                      /ACK
4500 002C 0016 4000 4011 B950 C0A8 0009 C0A8 0001 22B8 22B8 0018 EA03 C100
```

Upon setting `(l)ength:44` the report contains only the Ethernet monitoring head, IP datagram head, UDP head and the flags. The 4 reports correspond to communication over the Ethernet line. The third of them containing the data, after setting `l(e)ngth: 100` looks like:

```
>>O.K.
>>
12:58:51.829|eth:TX   68 |0002A94CA177 |0002A94C9D8F | IP/UDP/MOR/RET/DAT
4500 0036 009B 4000 4011 B8C1 C0A8 0001 C0A8 0009 22B8 22B8 0022 3B63 D200
1390 690F 8909 690F 8101 0A89 690F 8909 690F 8101 AAAA AA63
```

Structure of this report:

- 12:58:51.829|eth:TX 68 |0002A9 4CA177|0002A9 4C9D8F| IP/UDP/MOR/RET/DAT
  — head of the report
- 4500 0036 009B 4000 4011 B8C1 C0A8 0001 C0A8 0009
  — IP head
- 22B8 22B8 0022 3B63
  — UDP head
- D200 1390 690F 8909 690F 8101 0A89 690F 8909 690F 8101 AAAA AA63
  — MORSE pseudoframe

The content of individual report parts (guide description only):

Head of the report:

12:58:51.829	— time stamp
eth:TX	— Ethernet channel, transmitting
68	— 68_{dec} transmitted bytes in the monitoring
0002A9 4CA177	— destination Ethernet address
0002A9 4C9D8F	— source Ethernet address
IP/UDP/MOR/RET/DAT	content of the report (IP head / UDP head / Morse / retransmission / packet containing data)

IP head:

4500 –

4 – IP protocol version

5 – IP head length (5 words, each with 32 bits)

00 – type of desired service for datagram transmission

0036 — hex = 54 dec – length of IP datagram in bytes

009B — hex – serial number, datagram identification

4000 — 0000 = bits 0(most high) to 15(lowest):

fragmentation

- bit 0: 1= fragmentation forbidden
- bit 1: 1= next fragments follow
- bit 2: reserve
- bit 3 to 15: fragment offset – start position of fragment data part relative to original datagram (in bytes)

4011 — itemized:

40 — datagrams lifetime in seconds is decremented by at least 1 upon passage through the router at 1, (if =0 then discarded)

11 — specifies higher level protocol, the message of which is contained in the data part of the datagram

B8C1 — checksum of IP head

C0A8 0001 — source IP address

C0A8 0009 — destination IP address

UDP head + flags:

22B8 — source port UDP

22B8 — destination port UDP

0022 — 22 hex = 34 dec – length of UDP packet (UDP head + UDP data) in bytes

3B63 — UDP checksum

MORSE pseudoframe:

D200 — flags, here:

- `D` – data frame
- `2` – appended frame no, repeat no, security yes, problem bit no
- `0` – reserve
- `0` – protocol version

1390 — UDP link No

690F 8909 — to address

690F 8101 — from address

Morse packet contained in the MORSE pseudoframe:

0A89 — packet type

690F 8909 — destination address

690F 8101 — source address

AAAA — transmitted data

AA63 — checksum

18.4.2. ETH monitoring – CNI layer

This menu is accessible from MORSE main menu by command:

iMEle Enter Enter

```
    CNI Monitoring:
          |         retranslation         |            user              |
    s-moR |-RX--TX--ERR-len-type-mask-Spl-|-RX--TX--ERR-len-type-mask-Spl|
(0) 0 OFF | OFF OFF OFF   0 0000 0000 OFF | OFF OFF OFF   0 0000 0000 OFF|

de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
>>
```

The part `retranslation` is used for monitoring of traffic between the retranslation ETH input/output and the Node, i.e. at IP retranslation (M-IP-M).

The part `user` is used for packet tracking between ETH user input/output and the Node, i.e. at IP2Morse (IP-M-IP) or MAS (M-MAS-IP).

The following menus are similar to those for SCC channels:

Type **`0 Enter`**:

```
ETH CNI Monitoring:
(s)ys. channel:0
(r)etranslation
(u)ser
(d)st :00000000  (D)st mask :00000000 (1)-for :RX
sr(c) :00000000  sr(C) mask :00000000 (2)-for :RX
t(o)  :00000000  t(O) mask  :00000000 (3)-for :RX
fro(m):00000000  fro(M) mask:00000000 (4)-for :RX
use MO(R)SE addresses:OFF
(I)nit (W)rite
(q)uit
>>
```

The retranslation menu was chosen:

r Enter

```
Ret. CNI monitor:
O(N) O(F)F          (R)X:OFF  (T)X:OFF  TX (E)rr:OFF
eliminate (d)eadlock:ON  (S)plit:OFF
(l)ength:0
Packet (t)ype:0000  type (m)ask:0000
(q)uit
>>
```

For the meaning of items see SCC channel monitoring.

The format of a monitoring report for ETH – CNI layer

```
13:34:05.960|690F8909 690F8101|690F8909 690F8101|E00I    IN    2N89 1dat
AAAA
13:34:05.961|eth:TX   68 |0002A94CA177 |0002A94C9D8F | IP/UDP/MOR/RET/DAT
4500 0036 00A5 4000 4011 B8B7 C0A8 0001 C0A8 0009 22B8 22B8 0022 F982 D200
139A 690F 8909 690F 8101 0989 690F 8909 690F 8101 AAAA ED39
```

Packet `AAAA` in this example was incoming from the Node to the ETH channel (CNI layer monitoring) and was send to the Ethernet line (physical layer monitoring).

Meaning of items:

13:34:05.960	— time stamp
|690F8909 690F8101|690F8909 690F8101	— addresses in the order - to, from, destin, source
|E00I IN 2	— channel monitored, direction, length of the packet
|89 1dat	— packet type, serial number 0 to 7, kind of packet
AAAA	— user data

18.5. Monitoring NAG channels

The monitoring parameters of NAG channels are set here. Setting is performed individually for each channel.

From the MORSE main menu type **iMGIe Enter Enter:**

```
CNI Monitoring:
          |         retranslation         |            user             |
    s-moR |-RX--TX--ERR-len-type-mask-Spl-|-RX--TX--ERR-len-type-mask-Spl|
(0) 1 OFF | OFF OFF OFF   0 0000 0000 OFF | OFF OFF OFF  90 0000 0000 OFF|
(1) 0 OFF | OFF OFF OFF   0 0000 0000 OFF | OFF OFF OFF   0 0000 0000 OFF|

de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
>>
```

The **Network Agent** channels have not a physical input/output, so the menu offers the CNI monitoring only. The parameter setting is analogous like at SCC channels.

19. Statistic logs

From MORSE main menu select `d(i)ag stat (l)ogs`.

Type **il Enter**.

```
Logs:
(l)inks c(a)lls
SCC p(o)rts
net(u)ser
RF (m)edium
(e)thernet
pu(s)h into history list
clea(r)
(q)uit
>>
```

Menu commands common for all statistic logs:

pu(s)h into history list — All currently active logs (period 0) are closed, saved to the flash memory and cleared. The oldest logs history record (period 3) is overwritten, so the result of the `pu(s)h` command is that period 2 becomes period 3, 1 becomes 2, 0 becomes 1 and a new, empty record for period 0 is started.

clea(r) — Clears all currently active logs, i.e. period 0. The above commands should only be used in exceptional situations. When statistic information from the very last minute is needed, then use the commands `(d)isplay with (r)ef, reference (w)rite` see bellow.

Menu items `(l),(a),(o),(u),(m),(e)` enable selection of the desired log. To give an example, the `(l)inks log` has been chosen here:

Type **l Enter**

```
(0) (1) (2) (3)
(f)ilter
(d)isplay with (r)ef
reference (w)rite
(q)uit
>>
```

The above submenu is identical for all log types. The time period from which we want to see the statistics is chosen here. (`0, 1, 2, 3`) For logs consisting of list (`Link list a Call list`) a filter can be configured. Only list items with addresses meeting conditions set in the filter are then reported and displayed.

— For filter setting

Type **f Enter**

```
Log report filter:
(a)ddress:690F0300
(m)ask:FFFF0000
(o)ut:OFF
(c)nt:1
(q)uit
>>
```

(a)ddress: 690F0300 — comparative address (base)

(m)ask: FFFF0000
- `FFFF0000` — "1" bits indicate the part of the list item address which must be equal to the comparative address
- `00000000` — displays all items

(o)ut: OFF
- `OFF` — items where the masked address is equal to the base address are listed in the report
– `ON` — items where the masked address does not equal to base are listed

(c)nt: 1 - only this items are displayed, which have higher number of records than this parameter. Then `(c)nt:0` displays the the singly packet also.

Following commands can be used to get the latest (incremental) information from any statistic log:

```
(d)isplay with (r)ef
reference (w)rite
```

First it is necessary to get the reference log data. Type `(0)` to obtain the current log information, then use the reference `(w)rite` command to save it. After next typing of `(0)` you can display the increment of log data by issuing the `(r)ef` command. The `(d)isplay` command displays the current content of Setr buffers. Thus alternating the `(d)isplay` and `(r)ef` commands, you can have both, the incremental and the last received log data display.

(0),(1),(2),(3) — Choose the time period from which the log statistic is to be displayed

- `(0)` is the running period
- `(1)` is the last history period saved in the flash memory
- `(2)` the period before
- `(3)` the oldest history period available

The typical history period length is 86400 seconds, i.e. one day. The log period length can be configured within the `Unit menu`. The shift occurs also when you restarted CU, see pu(s)h into history list

Type **0 Enter**

The running period of the links log has been chosen - see the following chapter for the report.

19.1. LINKS LIST

Updated for version 8.04

This provides comprehensive information about the retranslation packets exchange. Contains the packets transmitted (TX) via the net N and link L node outputs and the packet received (RX). The log contains the communication of all nodes in CU.

From MORSE main menu select: `d(i)ag stat (l)ogs (l)inks (0)` i.e.

Type **ill0 Enter**

The log list starts with a header containing:

Log from unit	serial number of a tracked CU, decimal
history period	record period according to the menu `ill`
opened:	the time, when the log counters were cleared and the counting began
closed:	the time, when the counters were closed (for past periods) or the time, when the counters were read (for current period)
total	total counting time of displayed log in seconds for the choice `(0)` it is `(closed – opened)` for `(d)isplay` it is `(closed – opened)` in moment of reading by command `(0)` for `(r)ef` it si the time pertinent to the interval displayed
LINK LIST	distinguishing of the log type LINK LIST, CALL LIST,....

The remote address of a link listed in the report has to meet conditions set in the `(f)ilter` menu (see the `(f)ilter` menu)

```
>>
Log from unit 4833383  history period 0
  opened: 2007-02-05 13:29:59
  closed: 2007-02-05 13:44:33

total 874 s  LINK LIST
690F0001 690F0002
TX:  16/.018  byte:  3000/100  rep:  14/.467  lost:  8/.500  busy:  9/.360
RX:   8/.009  byte:   800/100  rep:   5/.385  misc:  0/ -
Total:
packs     bytes     packs/sec bytes/sec
24        3800      0.03      4.3
>>
```

Items of LINK LIST report

690F0001	node address of the link end in the observed CU, each node pair has a individual record in the log

690F0002 node address of the link end in the counterpart CU, in case of the wires retranslation the addresses are equal

Transmitting reports TX

TX:424/0.01 number of data packets sent into retransmission channel
more precisely:
packets entered in the RFC input buffer,
most 9 positions for MR400, next incoming packets are refused and registered in the busy item,
the TX packets are processed in some of next method:

- if sec OFF - packet is sent without requirement for ACK
- if sec ON - packet is sent and the ACK returned
- if sec ON - packet is ent and the ACK didn't come, the repeat follows and it is registered in the item rep
- if the ACK does not come after specified number of repeats, then the packet is discarded and registered once in TX and once more in the item lost
- if the ACK does not come until the store timeout in node expired (Ne1T15), then the packet is discarded and registered in the item lost
- if the store timeout expired and the packet did not start be transmitted, then the packet is discarded and it is not registered, neither in the TX item number of

TX: .../.018 `TX/total` — number of TX / total time of opened log, [packets/sec]

byte: 3000/... data byte number in TX + number of repeatedly transmitted bytes

byte: .../100 `byte/(TX+rep)` — average data lenght of packets transmitted

rep: 14/... number of repeatedly transmitted packets

rep: .../.467 `rep/(TX+rep)` — relative part of packet repeated

lost: 8/... number of packets which did not be acknowledged by ACK or which were discarded after store timeout,
the repeated packets (rep) are not counted
the unsecured packets are not counted

lost: .../.500 `lost/TX` — relative part of unacknowledged packets

busy: 9/... packets, which did not fit into buffer and were discarded

busy: .../.360 `busy/(TX+busy)` — relative number of refused packets

Receiving reports RX

RX: 8/... number of data packets received for the first time (the state of repeat bit in the header is not important here)

RX: .../.009	`RX/total` — number of packets received for the first time/sec
byte: 800/...	number of data bytes received for the first time
byte: .../100	`byte/RX` — average data length in the firstly received packets
rep: 5/...	repeatedly received packets, it is not added in the RX
rep: .../.385	`rep/(RX+rep)` — relative number of repeats
misc:	the meaning differs according to RF channel access mode:

- `NORMAL` - number of received packets having the data integrity mistakes (corrupted packets), this packets have the head crc OK but data crc wrong
- `R-CTRL` - number of tx req

Total - summary records for the all CU

packs 24	`TX+RX` — number of packets passed through CU, without repeats
bytes: 3800	`byte(TX)+byte(RX)` — general data flow, including repeatedly transmitted bytes, the bytes received repeatedly are not counted in
packs/sec: 0.03	`packs/total` — number of packets (TX+RX) / total time opening of log
bytes/sec: 4.3	`bytes/total` — number of bytes / total time opening of log

19.2. CALLS LIST

Offers a review of all the calls (MORSE addresses) which were detected on any of the RF Channels during the log period, together with average DQ and RSS values.

Fromn the MORSE main menu select: `d(i)ag stat (l)ogs c(a)lls (0)`, i.e.

Type **ila0 Enter**

Every address listed in the report has to meet conditions set in the `(f)ilter` menu.

```
Log from unit 4943628  history period 0
  opened: Thu Sep 30 12:29:59 2005
  closed: Thu Sep 30 19:03:03 2005
total 23584 s RF CALL LIST
addr.    count DQ RSS    addr.    count DQ RSS    addr.    count DQ RSS
69509001   766 18 109    690F0400   130 24  72    69501500    97 26  72
008A003A    77 23  72    690F8100    19 28  63    690F8606   185 28  90
690F8601    20 25  90    69509002     1  0 119
Total:
packs    rate
1295     0.05
>>
```

Items of an address record in the RF CALL LIST:

adr 69509001 — MORSE address

count 766 — number of transmissions detected

DQ 18 — average data quality

RSS 109 — average RF signal strength

packs 1295 — sum of all detected packet transmissions on the RF channel

rate 0.05 — packets per second average rate

19.3. SCC PORTS LOG

provides statistics of received and transmitted frames on the RS232 interface of SCCs (to and from a connected remote terminal device).

From the MORSE main menu select: `d(i)ag stat (l)ogs SCC (p)orts (0)`, i.e.

Type **ilo0 Enter**

```
Log from unit 5711764  history period 0
  opened: Mon Sep 20 23:08:42 2005
  closed: Tue Sep 21 05:34:05 2005
total 23123 s COM PORTS LOG
Port  TX   bytes  size  RX   bytes  size
  0      no TX data         no RX data
  1      no TX data         no RX data
  2   293  16180  55    287   3368  12
>>
```

Items of COM PORTS LOG report:

Port 2 — number of SCC

TX 293 — direction SCC → terminal, count of frames

- `bytes 16180` – count of bytes
- `size 55` – average length of frame

RX 287 — direction terminal → SCC, count of frames

- `bytes 3368` – count of bytes
- `size 12` – average length of frame

19.4. NET USER LOG

Gives information of packets passing through the user part of the Channel to Node interface, thus enabling evaluation of the network load generated by the user connected via the respective channel.

The SCC, Ethernet or Network Agent channels can be used fot user access in the MORSE network.

From MORSE main menu select: `d(i)ag stat (l)ogs net (u)ser (0)` i.e.

Type **ilu0 Enter**

```
Log from unit 4611616  history period 0
  opened: Tue Sep 13 14:09:04 2005
  closed: Tue Sep 13 15:26:14 2005
total 4630 s  USER INTERFACE LOG
Port  TX/period  bytes  size  RX/period  bytes  size
S00       no TX data              no RX data
S01       no TX data            20/231      80    4
S02       no TX data              no RX data
E00       no TX data              no RX data
G00       no TX data              no RX data
G01       no TX data              no RX data
>>
```

Items of USER INTERFACE LOG report

opened: — time when log counters were cleared and counting started

closed: — time when counters were closed and saved (for history periods) or time when log counters were read (for running period)

total 4630 s — entire time of counting (closed – opened)

TX — direction Channel → Node, count of packets

RX 20 — direction Node → Channel, count of packets

period — the average time interval between packets (sec)

bytes 80 — count of data bytes

size 4 — average length of packet data (bytes / packets)

19.5. RF MEDIUM LOG

Gives indications about time occupancy of the RF channel. Based on periodical sampling of the RF channel status and RSS level every 8 milliseconds, percentages of time consumed by data communication and by interference are calculated and displayed. If more detailed information is needed, RF medium monitoring can be used.

From MORSE main menu select: `d(i)ag stat (l)ogs RF(m)edium (0)`, i.e.

Type **ilm0 Enter**

```
Log from unit 4340141  history period 1
  opened: Wed Jan 23 21:13:14 2002
  closed: Thu Jan 24 21:13:14 2002
total 86400 s  RF MEDIUM LOG
```

```
comms total  7.97%  free 87.57%
noise:        comms:
 112  2.47%   101-    2.20%
 104  0.90%   85-100  1.71%
  96  0.96%   69-84   0.21%
  88  0.03%     -68   0.10%
  80  0.07%
  72  0.00%   **TX**  3.74%
  64  0.00%
  56  0.00%
  48  0.02%
No RSS info  0.00%
FDI 95.37%  CI 35.05%
>>
```

Items of RF MEDIUM LOG menu:

comms total 7.97% — sum of comms and TX

free 87.57% — free capacity of RF medium (noise below -116 dBm)

noise: 112 2.47% .... — percentages of time when noise signals with different RSS levels are received

comms: 101- 2.20% .... — percentages of time when MORSE data communication is received, sorted according the RSS level

TX 3.74% — percentage of time when RF modem was transmitting

No RSS info 0.00% — percentage of time when RSS measurement failed

FDI 95.37% — Free Data Index – percentage of the receiving time, when the channel is free, i.e. without `comms` nor `noise` (related to the time without transmitting)

CI 35.05% — Collision Index – it is meaningful at heavy traffic only (for `**TX**` 10 and more percent) – proportion of `noise` time to `comms` time with omission of the weak signals. This number illustrated, for which part of signals received the time collision occurs, i.e. they are evaluated like the noise

19.6. ETHERNET LOG

Information about traffic in Ethernet channel.

From MORSE main menu type **ile0 Enter**.

```
Log from unit 5863503  history period 0
  opened: Tue Sep 13 14:09:04 2005
  closed: Tue Sep 13 15:44:33 2005
total 5729 s  ETHERNET LOG
RX:
```

```
packs    :  0         packs/s :  0.00
bytes    :  0         bytes/s :  0.0
busy     :  0         packs/s :  0.0
collision:  0         packs/s :  0.0
overrun  :  0         packs/s :  0.0
crc error:  0         packs/s :  0.0
short fr.:  0         packs/s :  0.0
align err:  0         packs/s :  0.0
long fr. :  0         packs/s :  0.0
mcast    :  0         packs/s :  0.0
bcast    :  0         packs/s :  0.0
TX:
packs    :  0         packs/s :  0.00
bytes    :  0         bytes/s :  0.0
busy     :  0         packs/s :  0.0
carrier  :  0         packs/s :  0.0
underrun :  0         packs/s :  0.0
retry    :  0         packs/s :  0.0
tot. retr:  0         packs/s :  0.0
retr. lim:  0         packs/s :  0.0
late col.:  0         packs/s :  0.0
heart b. :  0         packs/s :  0.0
collision:  0         packs/s :  0.0
>>
```

The informations concerning the Ethernet channel traffic are available here.

- packs — the number of packets passed through Eth channel
- bytes — the number of bytes passed through Eth channel
- ostatní — next records serves for productions purposes

20. Events

There is one central Event log in an CU. When an outstanding event is detected, a standard event record is generated. The second part of this menu item e(r)ror is designated for obsolete error log versions.

From MORSE main menu type **i E Enter**.

```
Event log pages
(0) (1) (2) (3)...(F)
(c)lear forma(t) ckh(k) (s)tatus
(q)uit
>>
```

Meaning of items:

`(c)lear` — clears all events

`forma(t)` — formats the memory space and clears all events Both `(c)` and `(t)` commands are reserved for special situations and should not be used during normal maintenance operations

ckh(k) (s)tatus — for service purposes

`(0) (1) ...` — pages of Event list, the latest event record can be found at the top of page `(0)`

type **(0) Enter**

```
>>
Event log name0
2005-09-14 05:19:26| COLD START EVT 2000,00000044,02D10000
2005-09-14 05:19:15| PS SHUTDWN EVT 1999,00000000,00000000
2005-09-14 05:17:08| A:565098BA  test
2005-09-14 05:16:16| A:565098BA  put   MON SCC 3 O.K.
2005-09-14 05:16:16| A:565098BA  put   MON SCC 2 O.K.
2005-09-14 05:16:16| A:565098BA  put   MON SCC 1 O.K.
2005-09-14 05:16:16| A:565098BA  put   MON SCC 0 O.K.
2005-09-14 05:13:00| A:565098C6  put   UNIT 0 O.K.
2005-09-14 05:12:27| SUPPLY BACK EVT 1051,00000000,00000000
2005-09-14 05:11:47| SUPPLY DROP EVT 1050,00000001,00000000
2005-09-14 05:11:38| A:565098BA  write NODE 4 O.K.
2005-09-14 05:11:38| A:565098BA  write NODE 3 O.K.
2005-09-14 05:11:38| A:565098BA  write NODE 2 O.K.
2005-09-14 05:11:38| A:565098BA  write NODE 1 O.K.
2005-09-14 05:11:38| A:565098BA  write NODE 0 O.K.
```

The first part is the time when the event was detected, the second is a short description of it. The events recorded in example, chronologically from below:

A:565098BA write NODE 0 O.K. — 5 configuration structures written by command write

SUPPLY DROP EVT 1050	— change the supply to accumulator, MR400 detect this state from hw signal PI lead from the clamp MAIN PWR OFF on supply MS2000, MR25 from parameters in the menu Unit
SUPPLY BACK EVT 1051	— turning back to power box
A:565098C6 put UNIT 0 O.K.	— writing Unit edit by Init command done from the remote CU having address 565098C6
A:565098BA put MON SCC 0 O.K.	— writing in iMSPe menu by Init command
A:565098BA test	— control some of tests by command start, stop or report
PS SHUTDWN EVT 1999	— switching OFF the supply
COLD START EVT 2000	— switching ON the supply

21. Tests

All diagnostic tests can be controlled by three CU services – test start, test report and test stop.

These services are activated by commands `(s)tart r(e)port sto(p)`, which are present in the submenu of each test. The remaining menu items serve for editing of test parameters.

Anytime a test is successfully started, it has been started with parameters sent from `Setr.exe` together with the start command. Consequently, the only way how to change a parameter of a running test is to stop it, set the parameter and start the test again.

21.1. STATISTIC TEST

This enables complex testing of a path in the MORSE network.

From MORSE main menu type **i t t Enter**

```
Statistic (t)est:RX/TX
(N):1   (d):690F8606h
s(o)urce:-
d(a)ta:abcdefg
random data (l)ength:0byte
(r)epeat period:1000ms + (j)itter:200ms
Qualit(y) time constant:10s
RSS (m)easure:ON
sec(u)rity:OFF
(g)o on:OFF
(s)tart r(e)port sto(p)
>>
```

Items of statistic test menu:

`(t)est :RX/TX` — mode of test

- `(1) RX` — receiving path only
- `(2) TX` — transmitting path only
- `(3) RX/TX` — full statistic test of both path directions

Mode selection influences the rest of the menu, as items not relevant to the selected mode are not available for editing.

`(N) :1` — The number of the node where the test will run. The node is from the CU where the test start service is activated. Simultaneously, the node selected is the originating node of the tested path.

Note

It is possible to start a test on a node different from the one to which the Setr.exe is connected. Whenever we are connected to a node from some

CU, we can control (using `(s)tart`, `r(e)port` and `sto(p)` commands) any test on the remaining nodes from that CU.

`(d): 69601000h` — Address of Node, which is at the opposite end of the path to be tested.

`s(o)urce:-` — Only for RX mode. Address of the counterpart node (which has to be in the TX mode) transmitting test packets.

`d(a)ta: abcdefg` — Arbitrary data, which are then added to every test packet, can be set here.

type **a Enter**

```
Data:
/0 /1 /2 /3 /4 /5 /6 /7 /8 /9 /A /B /C /D /E /F  ASCII:
61 62 63 64 65 66 67                             abcdefg
Data: (a) ASCII (h)ex (r)andom (c)lear
(S)um
(q)uit
>>
```

`(a) ASCII` — type ASCII characters to fill in the data

`(h)ex` — type hexadecimal characters to fill in the data

`(r)andom` — choose random data length and random data will be generated

`(c)lear` — clears data set

`(S)um` — adds check sum to the transmitted data, according to RDS protocol rules, for production purposes

`random data (l)ength :0byte` — random data will be added to every test packet (a new set of random data for every packet)

`(r)epeat period: 1000ms` — period of test packet transmitting `(r)`

`(j)itter: 200ms` — a random part is added to every period `0-(j)`

`qualit(y) time constant: 10 s` — time constant used in calculations of immediate path quality (the packets coming in last 10 second are counted with bigger emphasis)

`RSS (m)easure: ON` — RSS and DQ value is added to the test packet on every RF link on the path

`sec(u)rity: OFF` — test packets are sent in non-secured mode (no ACK and no retransmissions on RF links)

`(g)o on:` —

- OFF – test packets are transmitted regularly with the `(r)epeat period` configured
- ON – test packets are transmitted immediately after receiving the response for the previous transmission, or after the `(r)epeat period` expires

`(s)tart` — starts the test

`r(e)port` — gets report from the running test

`sto(p)` — gets report and stops the running test

type **s Enter**

```
Test on Node 1 is running
```

Type **p Enter**

```
STest report  STOP!!!  RX_TX
source 690F8100  dest 690F8606  Node 1  security OFF
current sec:      44318       test sec: 35204
packets send:     35135       received: 34906
bytes per packet: 112         period:   1002.0 ms

PINGS   TX: 35135 RX: 34906 Lost: 229
min:    174    max:   320    aver:    201  qual:    235 ms
 160:  16726    192:  16913   224:    286   256:    981

       Path          DQ    RSS   HOM          Path          DQ    RSS  HOM
(690F8100-690F0300) 29.3  62.6  0.9   (690F0300-690F8606) 26.9  95.2 0.7
(690F8606-690F0300) 28.4  92.1  0.9   (690F0300-690F8100) 29.9  65.3 0.8


PER: 1:153.4    BER:      7.3e-06
breaks count: (length: count)
    1:   221       2:     4      8:     1
longest breaks: (units: sec)
    8: 20131      2:  2735      2: 19115      2: 34065
    2: 35062      1:   400      1:   487      1:   526
    1:   750      1:  1197      1:  1400      1:  1407
>>
```

Meaning of items of statistic test report:

`RX_TX` — test mode

`source 690F8100` — originating Node address

`dest 690F8606` — destination Node address

`Node 1`	— number of the originating Node within the source CU
`security OFF`	— security setting (see above)
`current sec:44318`	— time in seconds after the last cold start of source CU
`test sec:35204`	— duration of the test
`packets send:35135`	— number of packets transmitted
`received:34906`	— number of packets received
`bytes per packet:112`	— the average length of received packet
`period :1002.0 ms`	— the average period
`PINGS:`	— part of the test report dealing with response time statistic
`TX: 35135`	— number of transmitted packets
`RX: 34906`	— number of received packets
`Lost: 229`	— number of lost packets
`min: 174`	— the shortest response time
`max: 320`	— the longest response time
`aver: 201`	— the average response time
`qual: 235`	— path quality is a number calculated from response times, using a special algorithm with parameter `Qualit(y) time constant`
`160: 16726 192:`	— there were 16726 pings between 160 and 192 ms etc. – this shows the distribution of response times

RSS and DQ values for all RF links included in the tested path:

`Path (690F8100-690F0300)`	— RF link (from – to)
`DQ 29.3`	— *Data Quality* – average
`RSS 62.6`	— *Received Signal Strength* – average
`HOM 0.9`	— *RSS homogenity* value (calculated in a similar way to standard deviation)
`PER :1:153.4`	— *Packet Error Ratio* (the probability that a packet is lost or corrupted)
`BER :7.3e-06`	— *Bit Error Ratio* (the probability that any received bit differs from that originally transmitted, the value is calculated from PER)
	Next list displays the distribution of break lengths, where a break is understood to be a succession of lost packets; consequently break

	length is the length of the break measured in packets (the number of successive lost packets).
`breaks count: (length: count)`	• `break` – packet loss • `length` – break length measured in packets (number of consecutive lost packets) • `count` – number
`1: 221`	— one packet was lost 221×
`2: 4`	— two successive packets were lost 4×
`8: 1`	— from 8 to 11 successive packets were lost 1×, `Breaks count` is assorted into intervals, which creates a progression similar to geometrical one: 1, 2, 3, 4 – 5, 6 – 7, 8 – 11, 12 – 15, 16 – 23, 24 – 31, etc.
`longest breaks :(units: sec)`	–
`8: 20131 2: 2735 2: 19115 2: 34065`	— time in seconds (from the last cold start) of individual breaks, listed according to the size of the break. The 12 longest are entered
`delayed packets: 1`	— all packets with break longer than 32768, i.e. very substantive drop-out

21.2. BER TEST

The BER test is a special test for evaluating the bit error rate. For the most part, it is used for laboratory measurement of the modem on a given fixed link where it is not necessary to reveal as much data in the report as in the Statistic test. When measuring the radio link, it is best to use the Statistic test.

On one side of the link, the BER Test must be set for transmission `(t)est TX`; the other side (the side to be evaluated) must be set for receiving `(t)est RX`. The setting for other parameters is contained in the section Statistic test.

The Ber test runs a one way Statistic test (one side TX, the other RX). Only the report is arranged to display only the bit error rate. The Ber is calculated from lost packets and their length.

From MORSE main menu type **i t b Enter**.

Menu for transmitting CU:

```
Ber (t)est:TX
(N):1   (d):690F8100h
s(o)urce:-
d(a)ta:
random data (l)ength:10byte
(r)epeat period:1000ms + (j)itter:0ms
(s)tart r(e)port sto(p)
```

```
(q)uit
>>
```

Meaning of items:

`(N): 1` — determination of TX Node

`(d): 690F8606` — address of RX Node

`d(a)ta:` — and other items – see Section 21.1, “STATISTIC TEST”

Menu for receiving CU:

```
Ber (t)est:RX
(N):1   (d):690F8100h
s(o)urce:8606h
d(a)ta:-
random data (l)ength:-
(r)epeat period:- + (j)itter:-
(s)tart r(e)port sto(p)
(q)uit
>>
```

Meaning of item:

`(N): 1` — determination of RX node

`(d):690F8100` — not used

`s(o)urce: 8606h` — the lower word of TX node address (the upper word has to be equal to that of the RX node address)

Report from RX unit:

```
Ber report  only RX
source 00008606  dest 690F8100
packets received:  403         bytes per packet: 14
PER: 1:101.8    BER:      8.7e-05
>>
```

`PER :1:101.8` — probability of packet lost

`BER :8.7e-05` — probability of bit lost (counted from packet size)

21.3. SEND PACKET

This test is used to transmit a precisely defined packet. With this service, it is possible to send a packet of any type and data structure to the MORSE network.

MORSE main menu type **i t s Enter**

```
Send packet: (N):1   (d):690F0300h
(t)ype:0009h
s(o)urce:690F8100h
d(a)ta:abcd
random data (l)ength:0byte
(r)epeat period:1000ms + (j)itter:0ms
(s)tart r(e)port sto(p)
(q)uit
>>
```

`(N): 1` determination of the originating node `from`

`(d): 690F0300` — destination address

`(t)ype: 0009h` — type of packet

`s(o)urce: 690F8100h` — source address (can be different from the originating node address) `(N)`)

The other parameters follow the statistic test submenu. There is one speciality about the Send packet test – when the `(r)epeat period` is set to zero, the test starts by `(s)tart` command, sends just one packet according to parameters and immediately stops itself (a one "shot" operation). In this case, a service report limited to "O.K." comes as a response to the `(s)tart` command.

Normally, for periodical transmission, a response to the `r(e)port` command looks like this:

```
SPack report
source 690F8100  dest 300  Node 1
repeat period 1000 ms
data size 4 bytes
>>
```

21.4. PING TEST

The Ping test is a Statistic test but with the report arranged so that a table with the response-times is displayed. If the `RSS (m)easure` is ON, the RSS/DQ table of RF links is also displayed.

From MORSE main menu type **l t p Enter**

```
Ping: (N):1   (d):69601000h
sec(u)rity:OFF
(g)o on:OFF
RSS (m)easure:ON
(r)epeat period:1000ms + (j)itter:0ms
random data (l)ength:0byte
Qualit(Y):OFF
Qualit(y) time constant:10s
d(a)ta:
(s)tart r(e)port sto(p)
(q)uit
>>
```

`Qualit(Y):`
- `OFF` – whole report
- `ON` – report offers only the quality value

For the meaning of remaining items see Section 21.1, “STATISTIC TEST”

```
Ping report
Node 1  source 690F8100  dest 69601000  security OFF

PINGS   TX: 115   RX: 114   Lost: 1
min:    462    max:   1051   aver:    559  qual:    552 ms
 448:     22    512:     86    640:      1    896:      4
1024:      1
       Path          DQ    RSS   HOM          Path          DQ    RSS   HOM
(690F8100-690F0300) 29.2  64.9  0.8   (690F0300-69501500) 30.4  70.1  0.3
(69501000-69601000) 29.9  99.0  0.0   (69601000-69501000) 28.8  96.4  0.9
(69501500-690F0300) 22.8  70.4  0.8   (690F0300-690F8100) 29.4  66.9  0.6
>>
```

For the meaning of remaining items see statistical test report

21.5. ROUND TEST

The Round test simulates a polling-type user application i.e. calling to more CU from one centre, each of them separately. The test sends echo packets to selected group of destination Nodes. The lost packet probability and the average response time information are then available for every destination tested. This test is mostly used for network acceptance tests or as a network traffic generator, while the more detailed diagnostic information can be obtained from statistic logs from all participating CUs.

From MORSE main menu type **i t r Enter**

```
Round test:
(N):1  (d)
d(a)ta:
random data (l)ength:0byte
(r)epeat period:1000ms + (j)itter:0ms
sec(u)rity:OFF
(g)o on:OFF
(s)tart r(e)port sto(p)
(q)uit
>>
```

Meaning of items:

(N):1 — determining the originating Node

(d) — list of destinations

type **d Enter**

```
Rtest editor
(c)lear
e(x)ample
( 1)e 0300
( 2)e 0400
( 3)e 8606
(q)uit
>>
```

`(c)lear` — clears all lines

`e(x)ample` — example and help for the editor

`( 1)e 0300` — a line determining destination(s)

Only the lower word of the destination address can be configured in the editor, the upper word always equals the upper word of the originating Node address.

The editor line starts with one of the following letters:

- `i` – Means interval and should be followed by two addresses separated by a space. All addresses starting from the first one up to and including the second one are entered into the list.
- `e` – Means that the following addresses (separated by a space) are simply included in the list of polled destinations.

`d(a)ta:` — and other items – see the Section 21.1, “STATISTIC TEST”.

Example of the round test report:

```
RTest report
source 690F8100  total TX 205

Node 1  security OFF
[dest - TX/PER/ping]
 300h - 69/1:69.0/96   400h - 68/1:+INF/230
8606h - 68/1:1.0/-NAN
>>
```

Meaning of items:

`source 690F8100` — address of the originating Node

`total` — the total number of packets transmitted

`Node 1` — number of the originating Node

`security OFF` — test packets are transmitted as unsecured

`[dest-TX/PER/ping]`
- `dest` – the destination address

- `TX` – the number of transmitted packets to this destination
- `PER` – lost packet probability
- `ping` – the average response time in ms

In the example shown there was one lost packet to destination `690F0300`, `690F0400` was OK and `690F8606` was switched off.

21.6. MEMLOAD

The memload test is used for loading one of the SW modules (MORSE A, B, E, W) from the source CU to the destination one. Loading of a new firmware module into an CU is a complex process and should be reserved for trained personnel only. Attempts to remotely load SW modules made by an unqualified person can have disastrous consequences for the network. The following paragraphs deal only with the respective `Setr.exe` menu and do not completely describe the software loading process! For more information, please contact RACOM technical support engineers (see www.racom.eu[1]).

From MORSE main menu type **i t l Enter**

```
Memload:
(N):1   (d):690F8100h
(E)xternal flash:OFF
ma(x) sectors:1   (m)odule:A
User module: fi(r)st:00300000   (l)ast:0037FF80
(s)tart r(e)port sto(p)
go MORSE (A)/(W)
(i)nit
(f)ire (k)ill (c)ontinue
go MORSE (B)
(q)uit
>>
```

There are three different types of items in the memload menu:

1. Memload parameters, which have to be set before the test is initiated by the `(s)tart` command
2. `(s)tart`, `r(e)port` a `sto(p)` commands with standard behaviour
3. Memload commands, which can only be applied to a running test. These commands are used to proceed from one state to another during the loading process.

ad 1.) Memload parameters:

`(N) :1` — source Node in the CU

`(d) :690F8100h` — destination Node (in the CU where the selected SW module will be be loaded to)

[1] http://www.racom.eu

`(E)xternal flash:OFF` — when this switch is ON, the selected software module is loaded into the external flash, which has to be connected to the main processor bus (used in production only)

`ma(x) :1` — max. number of memory sectors per packet (for the purpose of memload, the memory sector size is set to 128 bytes, regardless of the actual flash sector size)

`(m)odule :A` — choose the SW module to be loaded, see submenu `module`

```
(E)  E  – load modul E, D, G or H according to source CU
(A)  A
(W)  W  – modul W or A must be in source CU
(B)  B
(u)  USER DEFINED
(q)uit
>>
```

`User module:` — arbitrarily selected interval from the CU memory

`fi(r)st: 00300000` — start address of the memory interval

`(l)ast: 0037FF80` — end address

ad 2.) Standard test commands: `(s)tart r(e)port sto(p)`

ad 3.) Memload commands:

`go MORSE (A)/(W)` — starts MORSE A (W) in the destination CU

`(i)nit` — activates the connection between the Node, where the memload test runs, and the destination node. It is necessary to perform `(i)nit` after any change of state on any side (start of the test, start of MORSE A/W in the destination CU, an error received etc.)

`(f)ire` — starts the loading

`(k)ill` — forces interruption of the loading

`(c)ontinue` — continues the loading process

`go MORSE (B)` — by starting MORSE B this command performs a cold start of the destination CU

The memload report provides complete information about the current status of the loading process:

```
This is Memload v1.03 response
  max. MF sectors per packet :4
  loading module: MORSE A
  target        : 690F5513h
  status :ready Check result: none
  begin  :  374000h
```

```
end    :  37D800h
current:  374000h
talking to MORSE E
timeout 12000
Time elapsed: 0msec
Transfer rate: nankbps ▶
```

Meaning of items:

`max. MF sectors per packet: 1`	— the number of 128 byte sectors currently transferred in a packet
`loading module:MORSE A`	— the software module being loaded
`target:690F8100h`	— the destination node address
`status:ready`	— the status of the loading process (error states are displayed here)
`begin: 374000h`	— the start address of the memory part being loaded
`end: 37D780h`	— the end address
`current: 374000h`	- the start address of the currently transferred sector
`talking to MORSE E`	— indication of the SW module running in the destination CU (this information can be invalid while in an error state)
`timeout 12000`	— current value of the timeout

The status reported can have the following values:

- `stop` – when the report was performed before initialisation of the memload channel
- `memfill error No` – an error state (followed by the error number)
- `data continuity error` – a mismatch in addresses of sectors transmitted and acknowledged has been detected
- `waiting for memfill response` – normal state during loading
- `all sectors are O.K.` – loading successfully completed
- `automated` – automatic memload runs
- `ready` – memload is prepared to start the loading process

`Memfill error No` can have these values:

- `1` MF_CODE_ERROR – safeguard code error detected by the destination CU
- `2` MF_RANGE_ERROR – attempt to enter an incorrect flash range
- `3` MF_BOUNDARY_ERROR – attempt to enter an incorrect boundary sector

- `4` MF_SIZE_ERROR – data delivered have a different size than the one stated in the header

Most error states can be resolved by the sequence of `(i)nit` a `(c)ontinue` memload commands.

22. Channel send

This diagnostic function resembles the Send packet test. Nevertheless it is not a test and it does not deal with packets. It works with the physical layer of communication through the SCC I/O interface. Arbitrarily created data frames can be sent out from that interface, or forced to be received. Forced data transmissions from inner SCP port can be initiated, too.

From MORSE main menu type **i n Enter**

```
Channel data send.
(d)estination: SCC0-TX
d(a)ta :abcdef
(r)epeat period :2000ms
(s)tart sto(p)
(q)uit
>>
```

type **d Enter**

Meaning of items:

`(d)estination: SCC0-TX` — interface and direction of the data frame SCC0-TX sending

```
Destination.
(1) SCC0-TX
(2) SCC1-TX
(3) SCC2-TX
(4) SCC3-TX
(a) SCC0-RX
(b) SCC1-RX
(c) SCC2-RX
(d) SCC3-RX
(0) SCP port
(q)uit
>>
```

`d(a)ta: abcdef` — assigning of sent data (abcdef), see Section 21.1, “STATISTIC TEST”

type **a Enter**

```
   Data:
   /0 /1 /2 /3 /4 /5 /6 /7 /8 /9 /A /B /C /D /E /F  ASCII:
   61 62 63 64 65 66                                abcdef

  Data: (a) ASCII (h)ex (r)andom (c)lear
  (S)um
  (q)uit
  >>q
```

Data frame definition is done in standard data editor, see Section 21.1, “STATISTIC TEST” description for details on page.

`(r)epeat period: 2000ms` — repeat period of data frame transmissions. When zero value is configured, a single transmission occurs immediately after the `(s)tart` command is issued

`(s)tart` — starts sending

`sto(p)` — stops sending

22.1. Enable RF(T)X

This function is used to disable or enable transmitting of the RF part of the CU. It influences the actual (RAM) value of the `RFTX disable` switch in RF modem parameters.

From MORSE main menu type `i T Enter`

```
RF packet transmitting:
(d)isable  (e)nable
(q)uit
>>
```

type **`d Enter`**

```
TX disable
Are you sure? (Y/N)
>>
```

press **Enter**

`(T)X off` — state of CU

The current state of `RFTX disable` parameter can be found by query `(s)ervice unit (s)tatus`, where the message `All transmitting disabled!` appears in case of disabled transmitting. If this message is missing then the transmitting is allowed.

23. Path editor

The Path editor is used for creating and editing the path buffer stored in the Setr.exe application. The contents of this buffer define the path to the node in the remote CU (Morse Communication Unit). By using command **!p** all service requests from Setr.exe will be directed to this CU. This enables remote configuring and diagnostics of such an CU that is temporarily or permanently unavailable for normal routing.

From MORSE main menu type **p Enter**.

```
Path32 editor
(c)lear
(h)elp
(1) ...
(q)uit
>>
```

`(c)lear` — clears whole path

`(h)elp` — instructions for editing path buffer

type **h Enter**

`Type 1m to add your own address`	– adds own address
`Type 1hAAAAAAAA to add 32bit address`	– adds defined address
`1d -delete`	– deletes address
`1cXXX... -creates address from previous one`	– create address from previous one creates new address from previous
`See ? in the main menu for additional help`	– ? from Main menu offers help

Example of how to enter path:

type **1m Enter**

type **2c0300 Enter Enter**

```
Path32 editor
(c)lear
(h)elp
( 1) 690F8100
( 2) 690F0300
(q)uit
>>
```

23.1. Path packets and communication with remote stations

The simplest way of making a connection to a remote station is by using command `!h` with the address of the opposite station, e.g. `!h690F0300`. The pathway from the initial CU to the destination CU and back can be either trivial or goes through another CUs which provide for store-and-forward retranslation. All CUs concerned must have appropriate entries in their respective routing tables. It is then possible to send packets to this address or carry out path tests using command `!`.

The same result can be achieved by compiling short paths in the Path editor with the initial address on the first line and the destination address on the second line. The path is activated using command `!p` and thereby the path is set-up.

Between the initial address and destination address it is possible to add one or more addresses. The path between each neighbouring address pair must then be defined in the retranslation tables, i.e. from `(1)` to `(2)` and back, from `(2)` to `(3)` and back etc. Each of these added addresses becomes the destination address for the relevant section of the path packet and according to this address the path is searched in the retranslation tables. Therefore it is possible to create a new path combined from several sections, where each of these sections is defined in routing tables.

The route from the initial to destination CU or some of its sections can be created from direct defined steps without the need to use retranslation tables. If so the whole of this section has to be described in detail, which means that all steps must be there. Each step is labelled in the path editor with a label, which is a row of zeros inserted before the step address. The step address is then set according to the same standards that are valid for addresses in the routing tables. During the passage of a packet through a Node a path defined in such a way appears as follows: address 00000000 is found at the place of the next destination address, which means that the address of the next step is not searched for in the tables, but the following address in the path table is used, i.e. the address of the direct defined step. The resulting entry of the relevant section in the Path editor is now made up from the initial and destination addresses of the section, between which are written address pairs from which the first is `00000000` and the second is the address of the direct step.

Example:

```
Path32 editor
(c)lear
(h)elp
( 1) 690F8100 starting address of the 1st section
( 2) 690F0300 target address of the 1st section,
              which is led on retranslation tables
( 3) 00000000 label 2.1
( 4) 690F8606 directly defined step 2.1
( 5) 00000000 label 2.2
( 6) 69501500 directly defined step 2.2
( 7) 69501500 target address of the 2nd section,
              which is created by the directly defined steps
( 8) 69501000 target address of the 3rd section,
              which is led on retranslation tables
(q)uit
```

>>!p – activation of path

`path 69501000h>!` — dispatch of path packet

```
u S02   690F8100    R01
29/ 87  690F0300    R01
25/ 84  690F8606    R01
29/ 92  69501500    S00
  S01   69501400     -
   -    69501401    S00
  S02   69501000   serd

 serd   69501000    S02
  S00   69501401     -
   -    69501400    S01
  S00   69501500    R01
26/ 96  690F8606    R01
26/ 90  690F0300    R01
30/ 77  690F8100  u~S02
path  69501000h>
```

With a `!pxxxx` command e.g. !p1401 it is possible to change the last address in the current path and thereby operatively change the destination of the path packet. This possibility is useful for creating variant paths, where their last section is found in the retranslation tables.

The Path packet in the whole listed range is functional from version 4.31.

24. Help

Under this item of menu are offered some possibilities, e.g. how to control a remote CU or how to work with path packets.

From MORSE main menu type **? Enter**

```
Shell commands:
!hXXXX - remote mode, destination XXXX(hex)
!dXXXX - remote mode, destination XXXX(dec)
!l     - local mode
!RSS (or ! itself) - Received Signal Strength service
!p - path
!pN - path C92
!pXXXXX - change last 1..4 bytes in current path to XXXXX
!pNXX   - dtto for path C92
>>
```

`!hXXXX` — the control is switched over to the remote CU with hexadecimal address `XXXX` (it is sufficient to type the right-most digits which differ from the previous address)

`!dXXXX` — the control is switched over to the remote CU with decimal address `XXXX`

`!l` — reverts the control back to the local CU

`!RSS or !` — after putting in the `!hXXXX` command you can by this command obtain the route listing with Received Signal Strength printout

`!p` — activates the path defined in `(p)ath` menu, it is possible to examine this path by `!` command

`!pN` — activates the path in C92 mode

`!pXXXXX` — changes last bytes in the last address in the current path (it is only used for the path, which has the table part at the end)

`!pNXX` — changes last bytes similar in the C92 mode

`!mfile` — starts macro `file.mac` in the running Setr

25. CNI-channel to node interface

This part of the menu is used to configure connections from an CU channel (of any type, i.e. SCC, RFC, ETH and NAG) to one of the nodes within the CU.

From MOSE main menu type **SIe** (or **FIe** or **EIe** or **GIe**) **Enter Enter**

```
 Channel to Node Interface:
    retranslation     |   user+service                lim
 id N  A t          m | N  A t Base     m  sec brc han S   e compr
(0) 0   NO AR         | 1   NO AR          usr OFF usr OFF NONE
(1) 0   NO AR         | 2 MASK 00000000/08 usr OFF usr OFF NONE
(2) 0   NO AR         | 1 MASK 00000000/08 usr OFF usr OFF NONE
(3) 0   NO AR         | 0 MASK 00000000/08 usr OFF usr OFF NONE

de(f)ault (r)ead (w)rite
(I)nit (S)ync
(q)uit
>>
```

The number of lines in the above table corresponds to the number of respective channels in the CU (4 for SCC or 5 for RFC or 1 for ETH or 2 for NAG). Choose No of channel to be edited:

type **1 Enter**

```
SCC Channel to Node Interface:
(r)etranslation
(u)ser+service
(I)nit (W)rite
(q)uit
>>
```

A channel can be used for user I/O or for a retranslation link (net), or for both, depending on the channel type and configuration. According to these two possibilities, the Channel to node interface is divided into two independent parts (retranslation, user). When the channel selected supports only one of these possibilities (e.g. an RFC cannot serve as a user I/O), settings of the other part are not used.

type **r Enter**

```
Retranslation CNI:
(N)ode:0
(A)R:NONE AR(t)No:0 (m)ask:0bits
(q)uit
>>
```

`Retranslation CNI:`

(N) (N)ode: 0 assigning a node to the channel retranslation output.

In the case of SCC during configuration of the link connection between two CU's and the chosen `(N)ode:1 - 4` the maintain packets are transmitted every 20 sec.

The transmission can be stopped by choosing `(N)ode:0`. See the MARS-A protocol for more informations.

(A) (A)R: NONE choose the way of address conversion

- `(n)NONE` — no conversion
- `(t)TBL` — by means of an address resolution table (ART)
- `(m)MASK` — by mask

(t) AR(t)No: 0 number of ART used

(m) (m)ask: 0bits - number of (least significant) bits used as the address mask

type **q Enter**

```
SCC Channel to Node Interface:
(r)etranslation
(u)ser+service
(I)nit (W)rite
(q)uit
>>
```

type **u Enter**

```
User CNI:
(N)ode:2
(A)R:MASK AR(t)No:0 (B)ase adr:00000000 (m)ask:8bits
user se(c):ON (s)ecurity:OFF (b)roadcast:OFF
(U)ser handicap:ON (h)andicap:OFF
limits: (S)econds:0s Byt(e)s:65535
z-(C)ompression:NONE
debug (I)NP:0 (O)UT:0
(q)uit
>>
```

`User CNI:`

(N) (N)ode: 2 - assigning a node to channel user output

(A) (A)R: BOTH - choose the way of address conversion

- `(n)NONE` — no conversion
- `(t)TBL` — by means of an address resolution table (ART)
- `(m)MASK` — by mask and base address
- `(b)BOTH` — both table and mask

(t) AR(t)No: 0 - if (A)R = TBL or BOTH, then number of ART

(B) (B)ase adr: 00000000 - if (A)R = MASK or BOTH, then base address, default address is 00000000

(m) (m)ask: 8bits - if (A)R = MASK or BOTH, then number of (least significant) bits used as the address mask

Security bit:

(c) user se(c):ON - the method of setting security bit in the packet incoming via this channel

- `ON` — the communication security in the channel controlled by the user packet type
- `OFF` – security according to the next `(s)ecurity`

(s) (s)ecurity:OFF - permanent setting of sec bit, for `(c):OFF` only

- ON — every packet sent must be confirmed
- OFF — non-secured communication on the communication channel

Broadcast bit:

(b) (b)roadcast:OFF - broadcast bit setting in the packet incomming via this channel

- `ON` – every packet entering the node is marked as a broadcast one
- `OFF` – packet goes through without change

Handicap bit:

(U) (U)ser handicap:ON - handicap bit setting in the packet incoming via this channel. This bit is utilized when solving the RF channel access, see the chapter RF Access.

- `ON` — bit handicap controlled by the packet incoming
- `OFF` – bit handicap set according to the next parameter `(h)andicap`

(h) (h)andicap:OFF - setting of handicap bit, for `(U):OFF` only

- `ON` – every incoming packet has set the handicap bit to 1, i.e. low priority
- `OFF` – bit handicap set to 0, i.e. high priority

limits: - limit for user data packets entering the network through this interface

(S) (S)econd: 10s - time limit (S = 0 means no limit)

(e) Byt(e)s :200 - length limit

(C) z-(C)ompression:NONE

- NONE — data go through without compression

- BASIC — in development - the data type user and prot are compressed on the input to MORSE network and decompressed on the output. This functionality allows the CU series MR400 and MC100 only. All CUs in the network must have set this parameter in the same way.

Index

A
address
  IP, 47
  MAC, 46
ARP table, locking, 48
ART
  MAS, 59
  resolution tables, 62

B
broadcast
  brc bit, 131
  IP multicast, 52
  routing, 32

C
carrier wave, transmission, 74
configuration parameters
  backup, 70
  savings, 9
  settings, 6

D
diagnostics, 79

E
ETH - ethernet channel
  address, mask, 47
  ARP parameters, 47
  mode M-IP-M, 54, 55
  mode MAS, 57
  pinger, 51

F
factory setting, 70
firmware
  upgrade, 119
  version, 72

H
help, 128

I
interconnection communication channels
  from channel to node, 129
  from node to channel, 21

L
log Events, 108

M
main menu, 5
MAS, 57
mobile mode
  new, 25
  old, 29
  old, state, 78
monitoring, 80
  ETH - CNI, 97
  ETH - physical, 93
  NAG - CNI, 98
  RFC - CNI, 91
  RFC - medium, 88
  RFC - physical, 87
  SCC - CNI, 85
  SCC - physical, 83
monitoring, format, 81
monitoring, sending, 81
monitoring, time, 81
mulicast IP frames, 52
multiaddressing, 21

N
NAG - network agent channel
  TMM tunnel MORSE MORSE, 60
noise, level, 73

P
path packet, 125

R
radio
  frequency setting, 18
  power setting, 19
restart sw, 77
RFC - radio channels
  access, 38
  mobile mode new, 25
  mobile mode old, 42
  priority, handicap bit, 131
  priority, parameters, 39
  transmitting disable, 124
  treshold DQ, RSS, 44
routing dynamic
  tables, 24

S
SCC - serial channels

idle, 36
protocols, 36
stop bit, 36
transmission speed, 35
serial number
modem, 11
radio, 17
sleep mode, 71
statistics, 99
communication with nodes, 101
ETH traffic, 106
filters, 99
receivable addresses, 103
RFC loading, 105
traffic on interfaces, 104
traffic on user interfaces, 104
supply, status, 71

T

tables resolution, ART, 62
tables routing
allocation, 22
content, 65
tables routing hierarchy, 31
tests
channel send, 123
memload, 119
send packet, 115
statistic, 110
time in CU
overview, 75
settings, 76
summer time, 13
synchronization, 13
transmitting disable, 43, 124

U

unit status, 74

A. Revision History

Revision 1.1 2007-01-30
The document title for MR400 series is SETR firmware[1] and for MR25 SETR firmware for MR25[2]

Hierarchy routing tables, DHe menu, new chapter

Revision 1.2 2007-03-30
RSS treshold, FMe o menu, new parameter

Revision 1.3 2007-09-18
New Mobile mode, DGe menu, new chapter, for more detail see MORSE Guide 2[3]

Frequencies for Mobile mode, Rpe 1 menu, new parameter

Next hop - dynamic choice of spare path, DNe menu, new chapter, for more detail see MORSE Guide 2[4]

Radio channel access, FAe menu, rewritten menu

Handicap bit sets the priority level for RFC access, SIe menu, new parameter

Revision 1.4 2007-08-22
Link list for fw 8.04, ill menu, updated

Revision 1.5 2007-10-23
IP-M-IP, address resolution, change of processing sequence (mask / Art)

Revision 1.6 2008-06-17
Noise level in sRs menu, specification

Treshold DQ for MR25 and treshold RSS for MR400

Diagnostic, d(i)ag menu, divided into more chapters

Revision 1.7 2008-07-15
Stop bit for SCC2+3 in MR400 series can take the values 1 or 2 only, see SPe menu

Revision 1.9 2008-08-27
M-IP-M, address resolution using mask, fragmentation

Revision 1.10 2008-12-12
Frequency step and channel bandwidth - specification, since 2008-12 labelled in the production code[5]

Revision History[6], list of changes in this manual since 2007-01-18 added

[1] https://www.racom.eu/eng/support/firmware.html
[2] https://www.racom.eu/eng/support/firmware.html
[3] https://www.racom.eu/eng/support/morse-m2/index.html
[4] https://www.racom.eu/eng/support/morse-m2/index.html
[5] http://www.racom.eu/eng/download/morsecode.html
[6] https://www.racom.eu/eng/support/firmware/mr25/apa.html

Revision 1.11 2009-01-05
Frequency step 6,25 kHz, indication in the Setr menu

Revision 1.12 2009-06-23
Art edit, routing tables redirect

Revision 1.13 2009-07-02
Sleep time interval enlarged (MR400 only)

Revision 1.14 2009-12-01
IP multicast, Art table extent

Revision 1.15 2009-12-04
ARP tables locking in NORMAL mode

Revision 1.16 2009-12-17
ARP tables locking in POSITIVE and NEGATIVE mode

Revision 1.17 2010-02-03
Index attached