

...the broadest narrowband money can buy



ADIO protocol for MORSE

version 10.0.62.0
2/28/2012

1. Introduction

The analog and digital input/output module for MR400 is named ADIO (Analog and Digital Input Output module) in this description. It is an optional module for MR400 radio modems. The module contains:

- 2 digital inputs (2.4 to 15 V for log 1)
- 2 digital outputs (30 V/500 mA)
- 2 analog inputs (0 to 20 mA or 4 to 20 mA)
- 2 analog outputs (0 to 20 mA or 4 to 20 mA)

Technical parameters see the MR400 manual¹.

The module works in the same way as the SEP technological unit and uses the same data format.

2. Data Format

The old format – SEP

Packet type in MORSE – `user data`

- 89 - secured or
- 09 - unsecured

SEP packet structure

`|DO/8|DI/8|8xAI/16|temp/16|2xAO/16|8xcount/32|crc/16|`

DO - digital output

DI - digital input

AI - analog input

temp - analog temperature of processor

AO - analog output

count - counter

¹ <http://www.racom.eu/eng/products/m/mr400/index.html>

Unused parts of the packet are filled with zeros. Counters are not used and their 8 double words also contain zeros.

The new format - MTF

Packet type in MORSE – `prot data`

- 8A - secured or
- 0A - unsecured

Structure of MTF packet

`|format/8|err/8|reqNo/8|respNo/8| data_part |chksum|`

`format` - 01 – first version

`err` - error message

`reqNo` - request number

`respNo` - response number

`data part` - the block containing commands and data, the number of `data_part` in one packet is arbitrary up to the maximal packet length 1400 Byte

`chksum` - the check sum

Data part:

`|type/8|ft/1|cmd/3|size/4|cnt/4|offset/12| data |`

(usual short `data_part` for SEP/ADIO)

`type` - type of `data_part` (read, write...) see MTF description

`ft` - format length of `data_part`

`cmd` - command

`size` - data block size in words

`cnt` - number of data blocks

`offset` - offset

`data` - the data, see the MTF description

The MTF is recommended for new applications.

Example of transmission of analog values in the MTF

Range 0 - 20 mA

			range	according to
5,010 mA	0x1392	input current, range 0 - 20 mA	0x0000 - 0x4E20	GPe 0pd TG
		Measured by the 12-bit A/D converter and converted according to the calibration constants		
	0x0401	Value sent to the network, 0 - 20 mA	0x000 - 0xFFFF	iMGle
			1 bit ~ 4,9 uA	
	0x0401	Value received from the network, 0 - 20 mA	0x000 - 0xFFFF	iMGle
		Converted according to calibration constants of the converter and converted to an analog value		
5,006 mA	0x138E	output current, range 0 - 20 mA	0x0000 - 0x4E20	GPe 0pd TG

Range 4 - 20 mA

			range	according to
5,010 mA	0x1392	input current, range 4 - 20 mA	0x0FA0 - 0x4E20	GPe 0pd TG
		Measured by the 12-bit A/D converter and converted according to the calibration constants		
	0x0102	Value sent to the network, 4 - 20 mA	0x000 - 0xFFFF	iMGle
			1 bit ~ 3,9 uA	
	0x0102	Value received from the network, 4 - 20 mA	0x000 - 0xFFFF	iMGle
		Converted according to calibration constants of the converter and converted to an analog value		
5,008 mA	0x1390	output current, range 4 - 20 mA	0x0FA0 - 0x4E20	GPe 0pd TG

3. Implementation in Morse

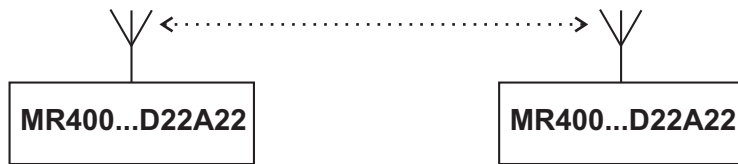
The ADIO module is used in a similar way to SEP, see the description of the SEP protocol: www.racom.eu, Products, MR400, MORSE, Detail description of individual protocols²

The ADIO module can be used in different ways:

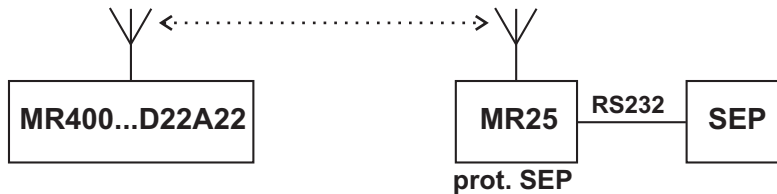
1. Connecting two MR400+ADIO over a radio link.

The state of inputs $D_{inp\ 0,1}$ and $A_{inp\ 0,1}$ is transferred to outputs $D_{out\ 0,1}$ and $A_{out\ 0,1}$ of the opposite station:

² <http://www.racom.eu/eng/support/protocol.html>

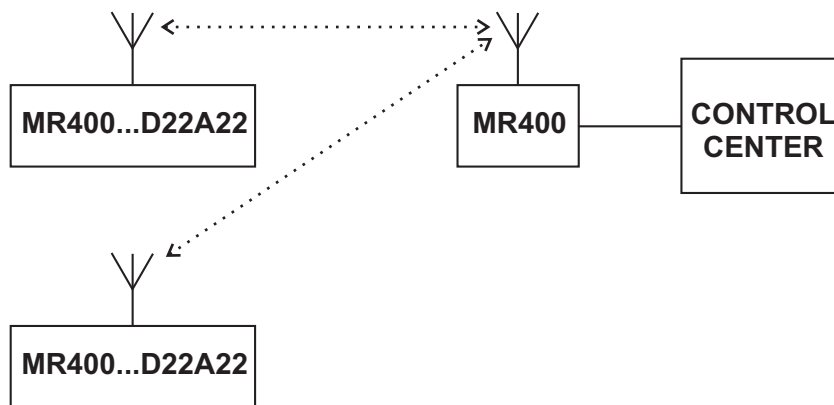


2. Connecting MR400+ADIO to MR25 and SEP:

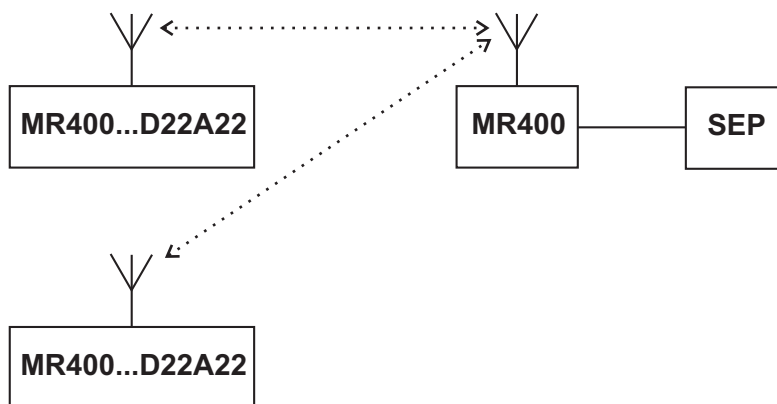


3. Connecting a greater number of ADIO modules to a control center.

ADIO modules are Slaves. The state of their inputs $D_{inp\ 0,1}$ and $A_{inp\ 0,1}$ is transferred to the center. Outputs $D_{out\ 0,1}$ and $A_{out\ 0,1}$ are controlled from the center. The MODBUS protocol compatible with MTF is used in the center.

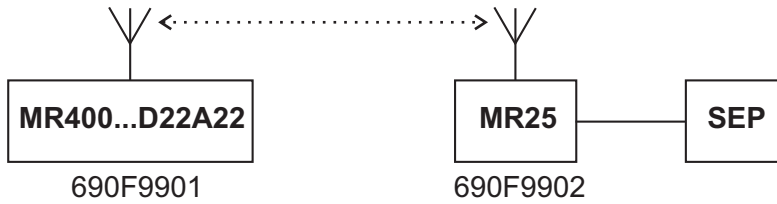


4. Connecting a greater number of ADIO modules to a central SEP in group mode (or two ADIO to one central ADIO module in group mode). Then to each I/O channel on SEP belongs one I/O channel in chosen ADIO module.



3.1. Example for the SEP format

Connecting MR400+ADIO to MR25 and SEP:



Packet format:

```
|DO/8|DI/8|8x AI/16|temp/16|2x AO/16|8x count/32|crc/16|
```

Monitoring RF channel on 690F9901:

```
10:34:43.754|690F9901 690F9902|690F9901 690F9902|0CA*31* 74 24|89 1dat
00FE 4AB5 0E80 0000 0000 0000 0000 0000 0000 0000 0000 4A6C 0E6E 0000 0070 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 7DDF
```

```
10:34:47.118|690F9902 690F9901|690F9902 690F9901|6A6 RFTX 20*09 0dat
FE02 4A9A 0E6E 0000 0000 0000 0000 0000 0000 0000 0000 4AB5 0E80 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 4D4F
```

The first packet is sent from the SEP PLC and byte `FE` describes the state of SEP inputs. It is copied to the ADIO output register and only two LSBs are physically set on the digital outputs. Similarly next word `4AB5` describes the state of analog input No.0 on SEP and is copied to an ADIO analog output.

```
00FE 4AB5 0E80 0000 0000 0000 0000 0000 0000 0000 0000 4A6C 0E6E 0000 0070 0000
: : AI0 AI1 A00 A01 counter00
: : (not used
: : in ADIO)
: ..DI FE = 1111 1110 = AI1=1, AI0=0
:DO 00 = 0000 0000
```

The second packet is sent later from ADIO. At the location of outputs this packet contains the original `FE` byte from SEP (actual state of ADIO digital outputs is only `AO1=1`, `AO0=0`). The following byte `02` describes the state of two digital inputs (`DI1=1`, `DI0=0`).

The eleventh word `4AB5` has a value obtained from SEP and represents the state of analog output 0 here. This output is connected by conductor to the AI0 in our example. This is why the size of the output current AO0 is the same as AI0 (19.10 mA and 19.12 mA, see below). The respective words are different (`4A9A` and `4AB5`). The cause of this is the different characteristics of the A/D converters, which are corrected by calibration constants.

The state of ADIO ports can be monitored with command "GPe 0pdT":

```
Test:
Set digi(0)ut Set (A)n out (G)et values
(q)uit
>>G
Gai0 19.10mA (4A9A) ai1 3.70mA (0E73)
ao0 19.12mA (4AB5) ao1 3.71mA (0E80)
di 0002/mask 0003
```

do 00FE/mask 0003 ... the mask reminds you that only the 2 LSBs are valid
>>

We can set up the digital outputs, for example, to state 0000 0011 using command "O3" (only the 2 LSB are set) or analog output to 15 mA using command "A1m15" or "A1 15000".

Next examples see the article SEP - examples³.

3.2. Example for the MTF format

A packet sent from the ADIO module to the MORSE network:

```
08:57:04.547|          |690F0003 00000001|G00I  OUT  42||8A 1usr  0
0100 01C4 01C3 1000 0003 0003 0001 06C3 1000 0003 0003 0000 02C1 2000 4000 0100
07C1 2000 4000 4000 C927
```

Packet giving a breakdown by the individual data_part:

```
08:57:04.547|          |690F0003 00000001|G00I  OUT  42||8A 1usr  0
0100 01C4          ... header
01C3 1000 0003 0003 0001 ... D input
06C3 1000 0003 0003 0000 ... D output
02C1 2000 4000 0100 ... A input
07C1 2000 4000 4000 ... A output
C927          ... checksum
```

More detail in Article MORSE technology format⁴

4. Configuration Parameters

Configuration is somewhat different to that of SEP and other protocols. SEP is a separate unit and a node communicates with it over the SCC channel using the SEP protocol.

ADIO is built into the CU and a node communicates with it via a Network Agent and does not need a special protocol. The Network Agent behaves in the same way as SCC in relation to the node, however if does not have an external hardware connection. Instead the Network Agents communicates directly with the ADIO module.

See the following example: We are using the command "Ne 1uG0":

Nodes:

```
retab
Nid|address |M | u   s | L   N |l w n g|sTO Err  Cent vTO hTO
(0) 00455774 - S00| - R00|0 0 0 0| 15 SERV OFF 304 30
(1) 690F9901 G00 S00| - R01|0 0 0 0| 15 SERV OFF 304 30
(2) 00000000 S01 S00| - R02|0 0 0 0| 15 SERV OFF 304 30
(3) 00000000 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
```

Also "GIe 0uAn" and "0uN1":

³ http://www.racom.eu/eng/support/protocols_docum/bc266__sepex.html

⁴ <http://www.racom.eu/eng/support/prot/mtf/index.html>

Channel to Node Interface:

retranslation		user+service		lim	
id	N A t	m	N A t Base	m sec	brc S e
(0)	0	NO AR	1 NO AR	ON	OFF NONE
(1)	0	NO AR	0 MASK 00000000/08	ON	OFF NONE

And now "GPe" and "0td":

Nag:

id	typ	subt	addr	mask
(0)	0003	0000	690F9902	001E0064
(1)	0000	0000	00000000	00000000

Using command "0pd" we obtain the parameters:

4.1. Configuration Parameters

The parameters are similar as for the SEP protocol:

Adio parameters:

```
(p)artner:690F0002 t(i)me:30s p(o)ll:10*10ms
n(e)t dead timeout:90s
analog (t)hr:200*10uA (f)ilter:0 (m)ask:3h
init (d)igi outs value:0h
init (a)na outs mode:all low value (0/4mA)
(N)egate digi outs mask:0h
Ala(r)m messages:OFF Alarm parameter(s)
(C)ounters:OFF (c)ounters parameters
(G)roup mode:OFF
(g)roup params
for(M)at:SEP
(A)out:4mA - 20mA
g(l)ue to resp:OFF
(D)ebug level:0
```

App m(O)de:normal

```
(1) power control params
(2) rfc usage alarm params
```

```
(T)est menu
(P)roduction
(q)uit
>>
```

(p)artner:690F9902 - MORSE address of opposite CU, 3 upper bytes are used for creating address in the (G)roup mode

t(i)me:30s - period for transmitting to (p)artner, if there were no input changes

p(o)ll:10*10ms - period Agent communicates with I/O module(100ms here)

n(e)t dead timeout:90s - if no packet arrives from the opposite station in this time, the output are set in the initial values

analog (t)hr:200*10uA	- packet is sent to the network upon exceeding this limit of the change on the analog input, max. value is 255*10uA CU also transmits upon each change of digital inputs.
(f)ilter:0	- low pass filter of changes on analog inputs, <ul style="list-style-type: none">• (f) : 0 - no filtering, the immediate values of analog changes are transmitted• (f) : 1 - to 10, larger values are not recommended – changes appear in a reduced value, the full change appears after a delay
(m)ask:3h	- digital inputs, whose changes should be transferred are marked with a bit mask 00-03h. At (m) : 3h reads both digital inputs.
init (d)igi outs value:0h	– Dout is set to this value after Init or start the protocol
init (a)na outs mode:all low value (0/4mA)	- both Aout are set to this value after Init or start the protocol
(N)egate digi outs mask:0h	– the masked digital channels are negated on the opposite CU
Ala(r)m mes- sages:OFF	- ON - joins a data_part with cmd=5 (alarm), if status defined in Alarm parameter(s) occurred, for MTF only
Alarm parameter(s)	<ul style="list-style-type: none">• defines the state of analog and digital inputs, at which is generated the Ala(r)m messages• Send (s)tatus info:ON - joins the data_part having type=05 (holding registers)
(C)ounters: OFF	- ON - in MTF joins the data_part having type=08 (counters) - obsolete
(c)ounters parameters	- obsolete
(G)roup mode:OFF	- selection of Group mode <ul style="list-style-type: none">• (o)n - commun. with addresses according to (g)roup params• o(f)f - communication with p(a)rtner address
(g)roup params	- Parameters are set individually for groups (0) and (1): Group processing: id 0 - first group parameters (t)hreshold:10*10uA remote (a)ddress:2 remote (g)roup:6 id 1 - second group parameters t(h)reshold:250*10uA remote a(d)dress:2 remote g(r)oup:7

(q)uit
>>

(t)hreshold:10*10uA - sensitivity threshold of analog inputs for causing transfer

remote (a)ddress:2h - lower byte of opposite station address, 3 upper bytes are taken from parameter (p)artner; address 00h disables the group from operation

remote (g)roup:6 - number of group in the opposite station

for(M)at:MTF

- (s) SEP - original SEP format for MORSE packet
- (m) MTF - new all-purpose format for technological MORSE packet data
- (b) SEP+MTF - sends the MTF format in the MORSE network, receives both SEP and MTF formats

(A)out:
4mA-20mA

- (4) 4mA - 20mA – current loop 4–20mA
- (0) 0mA - 20mA – current loop 0–20 mA

g(l)ue to
resp:OFF

- OFF - off
- ON - the glueing the spontaneous packet to command response possible

(D)ebug level:0

- 0 - without debug messages
- 1 - debug message is sent to the "ise" channel No.1

App m(O)de: - special modes

App mOde

- (0) normal - standard ADIO functionality
- (1) power control (solar) - mode(0) + low energy consuming mode
- (2) rfc usage alarm - mode(0) + indication of RF traffic level

(1) power control
params - parameters for low energy consuming mode (sleep)

Solar (A)ddress:00000000
S(u)pervisor address:00000000
A(s)leep after:0s
(set alarm time and mode in Unit menu!)

Solar (A)ddress - solar power supply MSU120 is connected to this node

S(u)pervisor addr - MORSE address of daemon, which switches off the CU

A(s)leep timeout - after this time the CU falls asleep, if no command arrives from the S(u)pervisor

(2) rfc usage
alarm params - parameters for indication of radio traffic level

Radio channel usage alarm:
 (drives DO0: alarm = negated initial value)

Ala(r)m refresh period:10s
 T(h)reshold:20%
 Noise (R)ss threshold:2

- Ala(r)m** - period 1–60sec in which the traffic is evaluated
- T(h)reshold** - the part of Ala(r)m period, when the radio signal is stronger than (R)ss threshold, e.g. 20%
- (R)ss** - level 1–10 describing how strong the RF signal must be to be evaluated, the scale according to "ilm" menu, see next table:

1	-112 dBm
2	-104 dBm
3	-96 dBm
4	-88 dBm
5	-80 dBm
6	-72 dBm
7	-64 dBm
8	-56 dBm
9	-48 dBm
10	stronger than -48 dBm

Function description:

The time occupation of RFC channel by the sum TX, RX and RX noise level stronger than Rss threshold is evaluated with the period of e.g. 10 sec. If the load exceeds the time limit chosen, e.g. 20%, DO0 is negated from its initial state "init (d)igi outs value:".

- (T)est menu** This menu allows you to set up values of digital and analog outputs and read the state of channels.

Test:
 Set digi(O)ut Set (A)n out (G)et values
 (q)uit
 >>

The following help appears after pressing O or A:

use `03' or `00x03' to set digital outputs or
 use `A(0|1) 2048' or `A(0|1) 0x800' or A(0|1)m19.3 to set analog output
 use `G' to get actual measured values
 >>

Examples:

We can switch on digital output DO1 using command:

```
>>O2
```

and set analog outputs AO0 to 7 mA and AO1 to 12.5mA:

```
>>A0 7000
```

```
>>A1 12500
```

We can read the resulting state using command "G":

```
>>G
>>ai0 6.983mA (1B47) ai1 12.450mA (30A2) ao0 7.000mA (1B58) ao1 12.500mA ►
(30D4)
di 0000/mask 0003 do 0002/mask 0003
>>
```

Item "do 0002/mask 0003" says that DO0=0, DO1=1. Mask 0003 reminds you that only the 2 LSBs are valid.

Inputs AI0 and AI1 are connected here to outputs AO0 and AO1 using conductors.

(P)roduction
menu

This menu is only designed for production purposes.

Pay particular attention in menu (P)roduction, "GPe 0pdP", which is used to set calibration constants. Safe commands in this menu are "G" or "E", which are used to read calibration constants or the serial numbers.

The conversion between the I/O value (0 - 20 000 μ A) and the A/D converter value (0 - FFF) is done according to the following formula

$$\text{I/O value}[\mu\text{A}] = k \times \text{A/D value}[\text{bit}] + q$$

Calibration constants k, q are computed and set in the Production menu.

This menu is only designed for production purposes, where the constants are set to the correct values. Before working in the Production menu we recommend reading the constants using command "(G)et" and making a note of them.

```
Production:
BEWARE! Changes calibration constants!
Calibrate analog.i(N)put (S)et output Calibrate analog o(U)utput
(G)et cal.constants Write (I)np cal.constants Write (O)ut cal.constants
G(E)t pid (R)ead eeprom Set (P)id
(q)uit
>>
```

Menu options:

```
(G)et          - reading calibration constants:
cal.con-
stants
```

```
>>G
ai calib.const k0 0518 q0 FFED k1 0519 q1 FFEA
ao calib.const k0 0308 q0 FFFE k1 0305 q1 FFFE
```

Write (I)np
cal.con-
stants

Write (O)ut - sets calibration constants, e.g. "ai calib.const k0 0518 q0
cal.con- FFED" can be set using command:
stants

```
I0 0x0518 0xFFED
```

If we set default constants, A/D converter values will then be used without conversion, e.g.:

```
I1 0x0100 0 - for analog input 1
```

```
O1 0x1000 0 - for analog output 1
```

Calibrate ana- - computation and setting of input calibration constants:
log.i(N)put First we apply an input current of 4mA to AI1 and write:

```
N1 4 Enter
0FA0 Cal.const set to default!
```

...if this message does not appear, repeat "N1 4"

```
>>resp: written x0 0FA0 y0 0315
```

...0x0FA0=4000dec, 0x0315=value of A/D converter

Now we apply the second known current (here 19.95 mA) to AI1

```
>>N1 19.95
4DED
>>x1 4DED y1 0F4D
```

...0x4DED=19950dec, 0x0F4D=value of A/D converter

```
Computed cal.const k0 5.098785(0519) q0 -22.941496(FFEA)
```

...computed constants

Relationships: $k = 5.098 \times 256 = 1305_{\text{dec}} = 0519_{\text{hex}}$, $q = -22_{\text{dec}} = \text{FFEA}_{\text{hex}}$

Enter calibration constants for AI1 to memory:

```
>>I1
0519 FFEA
>>resp: written
>>
```

Check result:

```
>>G
ai calib.const k0 0518 q0 FFED k1 0519 q1 FFEE
ao calib.const k0 0308 q0 FFFE k1 0305 q1 FFFE
```

(S)et out-put - setting output calibration constants:

(S)et out-put Calib- Set first output current to AO1:

```
>>S1 4
0333 Cal.const set to default!
>>resp: written resp: written
```

Measure actual current (here 4.35 mA) and write:

```
>>U1 4.35
x0 0333 y0 10FD
```

...0x0333=value of A/D converter, 0x10FD=4350dec

Set the second current value:

```
>>S1 18
0E65
>>resp: written
```

Measure the actual current (19.52 mA) and write:

```
>>U1 19.52
x1 0E65 y1 4C3F
```

...0x0E65=value of A/D converter, 0x4C3F=19520dec

Computed cal.const k1 0.188926(0305) q1 -2.637047(FFFE)

Relationships: $k = 0.1889 \times 4096 = 773\text{dec} = 0305\text{hex}$, $q = -2\text{dec} = \text{FF-FEhex}$

Enter calibration constants for AO1 to memory:

```
>>01
0305 FFFE
>>resp: written
>>
```

Check the result:

```
>>G
ai calib.const k0 0518 q0 FFED k1 0519 q1 FFEE
ao calib.const k0 0308 q0 FFFE k1 0305 q1 FFFE
```

G(E)t pid - reading the production number

```
>>  
Product: type 01 hwver 02 prod.num 000B date 081CA140 sign 12AB  
>>
```

(R)ead - for production purposes
eeprom

Set (P)id - for production purposes

5. Historie

Valid for the fw version 10.0.62.0