...the broadest narrowband money can buy

RACOM
RADIO DATA NETWORKS

MARS-A protocol for MORSE
(MARS rev.4)

version 7.45
10/10/2008

1. Introduction

MARS-A is a full duplex protocol with 32bit long addresses, with error detection (based on 16 bit checksum or 16 bit CRC) and with error correction. MARS-A could perhaps be characterised as the most user friendly of the network protocols in the MORSE system. With simplicity on the user interface, it offers access to all services in the MORSE data network system. Of course, the user has the option to choose whichever subsets to work with from the protocol.

The obsolete MARS-U (MARS rev.2) protocol (equipped with 16bit long address) is replaced by this new protocol. Longer address space is needed mainly for mobile applications in the MORSE network. The difference between this protocol and the obsolete mars-u protocol is that of longer address and different frame type.

2. Data Format

Data frame example in the MARS-A format:

```
|FT/2|FN/2|R/1|S/11| PT/8|H/1|L/1|R/3|N/3 |  A/32   | DATA |F/8|BCW/16|
|       C008       |  89 |      05        |690F 1244| AAAA |   | 98EC |
```

FT/2 frame type

FN/2 frame number

R/1 repeat

S/11 size, length of link layer data field, S= data length + 6 [byte]

PT/8 packet type

H/1 H2N bit
set to 1 if the transmitter is DTE=Host, A/32 is the destination address
set to 0 if the transmitter is DCE, A/32 is the source address

L/1 local mode

R/3 reserve

N/3 net number

A/32 destination or source address according to the parameter H/1

DATA user data, length is S - 6 [byte]

F/8 stuffing used to achieve an even size of the frame

BCW/16 checksum

The detailed description of the format follows.

2.1. The physical layer

The physical layer is given within the possibilities of the hardware in the MORSE system. In principle, MARS-A can operate in any arbitrary channel configuration. The standard assumption is for asynchronous communication between DTE-DCE on a duplex V.24 interface, using only data signals RXD, TXD, and ground (3-wire communication). Communication at higher speeds and at greater distances can use the RS422 interface. Synchronous communication is only possible on the V.24 interface.

2.2. The link layer

The link layer protocol can operate on an arbitrary physical interface enabling full duplex asynchronous communication with 8 data bits. It is fully transparent for data transfer, consequently flow-control Xon/Xoff cannot be used. It is also not necessary to use a hardware handshake as control of data flow is secured by the protocol itself. Due to its simplicity, this link layer protocol is useful for links with a negligible rate of error. The frame always has an even number of bytes. A set it is composed of a 16 bit label, a data block (always an even number of bytes), and a 16 bit control word (BCW).

There are three types of link layer frames:

- Data frame
- Control frame
- Service frame

Standard communication is Data frame - Control frame:

Example:
Data frame: `C009 0900 690F 8105 AB11 223A A828`
Control frame: `8106`
Answer:
Data frame: `E009 0900 690F 8106 CD33 442F 881C`
Control frame: `A106`

Data Frame

Data frame (including previous example):

```
| L/16 |       LLD/size         | F/8 | BCW/16 |
  C009   0900 690F 8105 AB11 22   3A     A828
```

L/16 Label.

LLD/size Link layer data.

F/8 Stuffing - byte with an arbitrary value used to achieve an even size of the frame. This supplementary byte is included in the BCW.

BCW/16 Block Control Word - Checksum mode is counted as lengthwise parity along 16 bit words in the whole frame (using XOR), default state. CRC-16 mode done according to RFC 1662 (the same as in PPP protocol) can be configured in the protocol parameters.

Label - 16 bits (1 Word) divided as follows:

```
| FT/2 | FN/2 | R/1 |     S/11     |
   11     00     0    000 0000 1001    ( = 0xC009 )
```

FT/2 Frame Type

- 11 - User Frame (normal mode).
- 10 - Control Frame (ACK...).
- 01 - Reserved for internal use.
- 00 - Link Layer Service Frame.

FN/2 Frame Number. Numbers are not incremented for repeated frames.

R/1 Repeat bit

- 0 - the frame transmitted for the first time
- 1 - repeated frame

S/11 Link layer data size (LLD), length of data field in bytes, not counting the label, stuffing and the BCW.

Control Frame

Control Frame:

```
| FT/2 | FN/2 | CC/4 |   CT/8    |
   10     00    0001   0000 0110    ( = 0x8106 )
```

FT/2 Frame Type

- 11 - User Frame (normal mode).
- 10 - Control Frame (ACK...).
- 01 - Reserved for internal use.
- 00 - Link Layer Service Frame.

FN/2 Frame Number. Equal to FN of acknowledged frame.

CC/4 Control frame class - currently 0x1

CT/8 Control frame type

- 0x06 - ACK - data frame was received correctly
- 0x05 - NAK - data frame was not received correctly invalid checksum, or size
- 0x04 - REJ - data frame was received correctly, but datagram sink is full

Service Frame

Time Get Service: read GMT from DCE

Request format:

```
| L/16 |0001/16| BCW/16 |
```

Response format:

```
| L/16 |0081/16|gmtsec/32|tfix/1|R/5|msec/10| BCW/16 |
```

Time Get Service: read GMT and localtime from DCE

Request format:

```
| L/16 |0002/16| BCW/16 |
```

Response format:

```
| L/16 | 0082/16 | gmtsec/32 | tfix/1 | ts/1 | R/4 | msec/10 |
| sec/8 | min/8 | hour/8 | day/8 | month/8 | year/8 | BCW/16 |
```

- Time information is valid when the first bit of the first byte of the frame is transmitted.
- The accuracy of time read is better, than 1 millisecond.
- When repeating, the time information is updated before sending the frame, thus a repeated frame will differ from the previous one.

Time Put Service: synchronise DCE's internal clock

Synchronisation command format:

```
| L/16 | 0003/16 | gmtsec/32 | R/6 | msec/10 | BCW/16 |
```

- Time information is valid when first bit of the first byte of the frame is received.
- The accuracy of time synchronisation is better, than 1 millisecond.

Confirmation format:

```
| L/16 | 0083/16 | BCW/16 |
```

Here:

L/16 label

Time GMT:

gmtsec/32 current time, in seconds,
elapsed since 00:00:00 GMT, January 1, 1970

msec/10 current milliseconds in running second, max. 999ms also 0x3E7

tfix/1 radio modem time fix quality

- 0 - radio modem time is well synchronized (according to it's configuration)
- 1 - radio modem time is not synchronized properly

ts/1 timesavings (1 - on)

R/4, R/5, R/6 reserved, must be zero

Local time (includes time zone and daylight savings):

sec/8 Seconds

min/8 Minutes

hour/8 Hour (0–23)

mday/8 Day of month (1–31)

month/8 Month (0–11)

year/8 Year (calendar year minus 1900)

2.3. Network layer

The network layer contains the network header and the data, that is in the previous example:

```
| NETWORK HEADER  |  DATA  |
 0900 690F 8105    AB11 22
```

Header structure (including previous example):

```
|   PT/8  | H/1 | L/1 | R/3 | N/3 |    A/32   |
 0000 1001   0     0    000   000  0x690F 8105  ( =0x0900 690F 8105)
```

Network layer data:

```
|  DATA  |
  AB11 22
```

PT/8 Packet type, see next chapter

H/1 H2N bit
set to 1 if the transmitter is DTE=Host, A/32 is the destination address
set to 0 if the transmitter is DCE, A/32 is the source address

L/1 Local
0 - normal

1 - local mode, setr command !l, A/32 recommended to be zero

R/3 Reserved, must be zero

N/3 Network number (transferred over the network)

A/32 Address

- Packet numbering is not compulsory. Only the lowest three bits from the number are transmitted. The entered number is transported to the address, and can serve to control the order of the delivered packets (the MORSE network does not guarantee a perfect order for packets).
- The address is the source address (Host receives a packet) or the destination address (Host transmits a packet). The packet goes in the CU through the CNI (Channel to Node Interface), which maps the user address space to the MORSE address space e.g. by a mask.
- The maximum allowable length of data in the MORSE network layer is 1626 bytes. Longer packets are not defined within the system. The optimum packet size for MORSE system is 200-400 bytes.

2.4. Packet Types

The Packet type composition:

```
|U/1|B/1|R/1|subt/5|
```

U/1 link security bit

B/1 broadcast (multicast) bit

R/1 reserved

subt/5 subtype

The subtypes of packets can be divided into 5 groups:

1. User packets
2. Request for services
3. Service reports
4. System reports
5. Special packets

Here we describe the most important MORSE packet subtypes:

09h - USER DATA The basic type of packet for transporting data from source to destination.

0Ah - PROT DATA This type of data is designed for data flow control in the user protocol. The service of both preceding types of packets in the MORSE network is equivalent. Packets are sent to the destination with the path and priority settings at the participants addresses. An error message is sent to the original sender if a packet is lost, but the packet carrying this error message can of course be lost as well, and in this case silently.

10h - SERVICE REQUEST	Request for a MORSE service.
12h - SERVICE REPORT	Report from a MORSE service.
0Ch - PACK ERROR REPORT	MORSE error message. First word is the Error Number, the rest of the message holds more detailed information about the network error. Generation of these errors can be turned off and on for the whole network. Some of these Error Numbers are here: • 1 - PACKET_NOT_CONFIRMED • 2 - STORE_TIMEOUT • 3 - NO_CHANNEL_ASSIGNED • 4 - ACCESS_TIMEOUT_ERROR • 6 - WRONG_PACKET_FORMAT • 7 - DEST_PROT_MISSING • 8 - WRONG_PATH • 9 - WIRE_LINK_FAIL

2.5. Byte order

All 2-byte and 4-byte values are transferred in normal order. The highest byte is transmitted first, and then the lower ones (beware, as the Intel processor format is typically the opposite).

3. Implementation in MORSE

3.1. The packet structure

The packet used in the next example has this composition:

```
F008 0980 690F 8902 AAAA B32F
```

This is more exactly (see Data format chapter):

```
F     0     0    8    0    9    8    0     690F 8902 AAAA B32F    bytes
11 11 0 000 0000 1000 0000 1001 1 0000 000                         bits

__ __ _ ____________ _________ _ ____ ___ _________ ____ ____
FT FN R    size          PT     H L,res  N  address  data  BCW    meaning
          (8 bytes)  |<--                            -->|
```

Here is H=1, so the address is considered as destination.

3.2. Input and output of the packet through MARS-A protocol

In this example the packet in MARS-A format is received via SCC2 port in CU 690F8901, goes through the MORSE network and then outputs from CU 690F8902 via SCC3 port. The monitored points are labeled *

```
      *          *                                          *
rxsim -> SCC0    ->  690F8901      ->    690F8902     ->  SCC1 -> tx
        MARS-A       source              destination      MARS-A
        protocol     address             address          protocol
```

```
>>
12:29:37.101 rxsim  12 | S02
F008 8980 690F 8902 AAAA 332F

12:29:37.102 tx      2 | S02
B106

CNI mon     |toa     frm    |dst      src     |           size|TT N
12:29:37.102|               |690F8902 690F8901|S02I   OUT    2||89 0usr 0
AAAA

12:29:37.135 tx     12 | S03
D008 8900 690F 8901 AAAA 13AC

12:29:38.143 tx     12 | S03
D808 8900 690F 8901 AAAA 1BAC
```

The monitoring contains 5 parts:

1. The inputting packet, H=1 and destination address.
2. Confirmation ACK on SCC2.
3. The packet outgoing from SCC2 to the node 690F8901, data only.
4. The packet transmitted from SCC3 port, H=0 and source address.
5. The repeatedly transmitted packet from SCC3, because the ACK on SCC3 is missing.

3.3. Next examples.

In the next example we send a request for the software version in a radio modem and we receive answers. Packet type is `10h` or `12h`, request for system software version is formulated as three bytes E0277600 (or E027 followed by the binary coded software version in the case of the answer). Address of the radio modem `690F0501` serves here as destination and source also. The protocol is configured in checksum mode.

```
//Send & receive
16:51:55.87|tx S0   14
D00A 1080 690F 0501 E027 7600 3AA3
16:51:55.87|rx S0    2
9106
16:51:55.88|rx S0   36
E020 1200 690F 0501 E027 0000 0045 01B2 0000 0000 0041 01B2 0000 0000 0042
01B2 0000 7FFD
16:51:55.90|tx S0    2
A106
```

```
//Send & receive with ACK after the frame
16:52:11.42|tx S0   14
E00A 1081 690F 0501 E027 7600 0AA2
16:52:11.42|rx S0   36
F020 1201 690F 0501 E027 0000 0045 01B2 0000 0000 0041 01B2 0000 0000 0042
01B2 0000 6FFC
16:52:11.43|rx S0    2
A106
16:52:11.45|tx S0    2
B106

//Send & receive
16:52:12.44|tx S0   14
F00A 1082 690F 0501 E027 7600 1AA1
16:52:12.45|rx S0    2
B106
16:52:12.45|rx S0   36
C020 1202 690F 0501 E027 0000 0045 01B2 0000 0000 0041 01B2 0000 0000 0042
01B2 0000 5FFF
16:52:12.48|tx S0    2
8106

//Send & receive with repeats
16:52:20.89|tx S0   14
C00A 1083 690F 0501 E027 7600 2AA0
16:52:21.99|tx S0   14
C80A 1083 690F 0501 E027 7600 22A0
16:52:23.09|tx S0   14
16:52:23.09|rx S0    2
8106
16:52:23.11|rx S0   36
D020 1203 690F 0501 E027 0000 0045 01B2 0000 0000 0041 01B2 0000 0000 0042
01B2 0000 4FFE
16:52:23.14|tx S0    2
9106

//Send with no answer
16:52:33.04|tx S0   14
D00A 1084 690F 0501 E027 7600 3AA7
16:52:34.13|tx S0   14
D80A 1084 690F 0501 E027 7600 32A7
16:52:35.22|tx S0   14
D80A 1084 690F 0501 E027 7600 32A7
16:52:36.32|tx S0   14
D80A 1084 690F 0501 E027 7600 32A7

//(MORSE error message should be generated...)
```

3.4. Link layer service example

Time set and time get example (via link layer of the protocol).

```
PLC sends the synchronisation command which sets the time in the CU,
GMT:17.8.2007, 08:29:35.873
monitored on SCC2 of synchronised CU,
the CU time was changed from 08:35:24 to 08:29:26
>>
08:35:24.870 rx;i   12 | S02
0008 0003 46C5 4E56 03E7 0B7F
08:29:26.992 tx      6 | S02
0002 0083 0081
08:29:26.992 tx      2 | S02
8106
08:29:27.002 rx;i    2 | S02
8106


PLC connected via MARS-A enquire about the time,
the CU responds using the longer format,
GMT:17.8.2007, 08:29:35.873
monitored on SCC2 of questioned CU:
>>
08:29:35.873 rx;i    6 | S02
0002 0002 0000
08:29:35.874 tx     18 | S02
100E 0082 46C5 4E5F 436A 231D 0811 076B 771B
08:29:35.874 tx      2 | S02
8106
08:29:35.890 rx;i    2 | S02
9106


PLC connected via MARS-A enquire about the time, the CU responds,
GMT:17.8.2007, 08:32:21.232
monitored on SCC2 of questioned CU:
>>
08:32:21.232 rx;i    6 | S02
0002 0001 0003
08:32:21.232 tx     12 | S02
2008 0081 46C5 4F05 40E8 69A1
08:32:21.232 tx      2 | S02
8106
08:32:21.245 rx;i    2 | S02
A106
```

Code fragment in C language, which will convert binary GMT to ASCII.

```
void main()
{
  time_t t=0x388BAD40;
  struct tm *gmt;
  gmt = gmtime(&t);
  console.printf("GMT is: %s", asctime(gmt));
}
```

3.5. Support for dial-up on GSM or telephone modems

The transfer is designated to occasional use, e.g. for the configuration changings in the distant communication unit (CU).

HW arrangement and the parameter setting is similar, like at the connection with async. link, see the example[1] in the MORSE Guide 1. The full RS232 cable for connection between SCC and the phone modem is recomended. In the case of using 3-wires cable connect the DTR and DSR pins on the phone modem. In the telephone modem, there is necessary set the parameter AT&K0 (the data flow control off) and ATQ0, ATV1. The autoanswer function can be ON, however it is executed from CU also. Considering the various characteristic of the phone modems it is necessary to test the configuration in advance.

The speed 19200 bit/sec is sufficient for fixed lines (can be to 57600), for GSM set max 9600. It should be in any case equal to the speed of telephone modem.

With help of `SPe 0t` set the parameters of protocol MARS-A on both CU equal:

```
MARS-A parameters:
(a):1000ms  (r):5  (c)rc:OFF  (G)SM:1  no traffic t(i)meout:0sec
(t)el.:
Opposite retranslation address:00000000
(l)oc:OFF  loc (s)ource:00000000
(R)emote dial-up :OFF   DO ORDER SPECIAL CABLE! DO SET IDLES TO 30!
(q)uit
>>
```

(G)SM:1 in calling CU activates the function of phone connection, in called CU activates AA (autoanswer)

t(i)meout:30 switches out the phone connection at longer dellay, than 30 seconds

It can be set either to 0 (the function is off) or the time longer then 30 sec, because after beginning of connection the holding packets are send every 20 sec and this at normal circumstances prevents the termination of connection.

STARTING OF CONNECTION:

Fulfil the telephone number `(t)el.` : to the parameters MARS-A protocol in the same form, like at telephone call:

```
        25  link inside the firm net
 566618578  public line
0566618578  call from the firm net "through zero"
```

Don't save this parameter by `(w)rite` command, but initialize it only:

```
(q)uit Enter
(I)nit Enter
```

The telephone modem makes the connection, which is visible in this way:

[1] https://www.racom.eu/eng/support/morse-m1/routex3.html

- the LED named OH (Off Hook) is on, the modem started work
- the modem dial the number (audible)
- if the start of connection with the opposite modem was succesful, takes place the connection parameter negotiation, which looks like the variable tone for pair of seconds. After succesful parameter negotiation the LED CD (Carrie Detect) is on
- from this moment it is possible to comunicate between both radio modems through phone line, e.g. send the ! command and change the configuration in the opposite CU

After the automatical sending of the holding packet, the address of opposite CU appears in the MARS-A parameters like “Opposite retranslation address”. When we need cancel the connection, we do Init with empty string `(t)el.:` in MARS-A parameters.

Check the cancel of connection, which shows like switching off CD and OH. In case of any problem we stop the connection by phone modem switching off and on. We choose from main menu `SPe0t` and check, if the parameter `(t)el.:` is empty. In other case the phone number would be dialled at next start of CU.

3.6. (R)emote dial-up access

This remote access via the telephone modem replaces the connection using the service cable. Configuration:

```
PC -- TM ---telephone network --- TM -spec.cable- CU
```

PC — PC with OS Linux and program `rr_setrdial`

TM — telephone modem

spec.cable — RS232 crossed cable, in which pin CD is connected to pin DTR at the opposite end of the cable

CU — MR400 series CU, fw 746 or higher, default configuration with the exception of parameter
`(R)emote dial-up :ON`

A remote CU is connected via the serial port to the telephone modem using a "Special dial-up cable". The CU can be set up with default configuration, the SCC used has protocol MARS-A set up with parameter `(R)emote dial-up:ON` . It is a good idea to set up the parameter idle to a higher value, for example
`SPe 2i  RX (i)dle:100`.

Set up parameters containing the telephone number and port in program `rr_setrdial`. After starting the program sets up a connection with the destination TM and then starts `rr_setr`.

After connection TM-spec.cable-CU the CU starts to send on wires each 30 sec the ASCII command autoanswer ATS0=1. The TM sets the communication speed according to CU. When PC through rr_setrdial calls the distant TM, then TM accepts the call, sets CD=1 and in this way appears DTR=1 on CU. Only now at DTR=1, the CU can send data via it's TM into distant application rr_setr.

Now we can work with Setr in the destination CU in the same way as if connected via the service cable. We are not connected via SCC0, as is the case with the service cable, but via SCC used for "special dial-up cable". Connection of the cable:

```
SCC              25-pin male
(radiomodem)    (dial-up modem)

1 - CD    ---   20 - DTR
2 - RxD   ---    2 - TxD
3 - TxD   ---    3 - RxD
4 - DTR   ---    8 - CD
5 - GND   ---    7 - GND
6 -              6 - DSR
7 - RTS   ---    5 - CTS
8 - CTS   ---    4 - RTS
9 -              9 - NC - NOT CONNECTED
```

4. Protocol parameter settings

MARS-A parameters:

```
MARS-A parameters:
(a):1000ms  (r):5  (c)rc:OFF  (G)SM:0  no traffic t(i)meout:0sec
(t)el.:
Opposite retranslation address:00000000
(l)oc:OFF  loc (s)ource:00000000
(R)emote dial-up :OFF   DO ORDER SPECIAL CABLE! DO SET IDLES TO 30!
(q)uit
>>
```

The link layer protocol is always configured using the protocol parameters.

(a) ACK timeout in milliseconds

(r) No of repeats (not counting the first trial)

(c) OFF = lengthwise parity used, default state

ON = CRC-16 mode

(G) telephone modem connection, see chapter Section 3.5, “Support for dial-up on GSM or telephone modems”

(i) switches out the phone connection automatically

(t) called telephone number

- Opposite retranslation address:00000000
  This is used in the store and forward relaying mode when two CU are connected by wires, the opposite CU address appears automatically after establishing the connection.

(l) OFF = normal state

ON = local retranslation mode for the service purposes, the communication with a remote unconfigured CU via the MARS-A protocols, see the MORSE Guide 2, chapter Local mode connection[2]

(R) OFF = normal state

ON = (R)emote dial-up, the service access into a remote CU via a telephone network, see the chapter Section 3.6, “(R)emote dial-up access”

There are two basic parameters:

- (a) - ACK timeout - the time after transmitting a data frame in which the transmitting station waits for an ACK frame. If the ACK does not arrive, the frame is repeated.
- (r) - number of repeats - if the number of repeats is exhausted and the ACK timeout expires, the network layer is informed that the packet is lost.

The state diagram on the transmitting side of the link layer protocol has only two states:

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

The “quiet” state, and the “waiting” state while expecting an ACK frame. Immediately after receiving a demand to transmit a data frame, the “quiet” state passes into the “waiting” state in order to receive the ACK. After receiving the ACK, or if the timeout expires after the last repeat, the link layer protocol goes back to the quiet state. While in the waiting state, the protocol declines all requests to send a frame.

The receiving side reacts instantly after receiving a correct data frame by sending an ACK frame. For a correct frame, the ACK is sent immediately after it is finished, so the ACK timeout must be set at least equal to the time for transmitting the longest frame. An incorrect frame is silently discarded.

Between two characters in the frame, there cannot be a greater delay than is set in the idle parameter. A split frame is evaluated as an error, even if this occurred due to the hardware handshake.

5. MARS Protocol History

revision 0 (MARS M) 3/1999	two 2 byte addresses, data frame type 00, size in net header, unnumbered ACK
revision 1 (MARS U) 4/1999	one 2 byte address, data frame type 01, (switch old on) unnumbered ACK
revision 2 (MARS U) 4/1999	one 2 byte address, data frame type 01, numbered ACK
revision 3 (MARS A) 12/1999	one 4 byte address, data frame type 11, numbered ACK,added REJ,NAK,added DTE/DCE field, time synchronization in link layer
revision 4 (MARS A) 9/2001	added broadcast bit and security bit to the packet type