

ELIUM RS232 Protocol Description

1. General

Date: 24.10.2007

Revision: 2.0

Model: DVB Signum Receiver

Scope: The goal of this document is to describe how ELIUM Receiver can be controlled through RS232 connector (RS-RC mode).

2. The RS232 attachment

One of the many features implemented in ELIUM Receiver application is the possibility of bidirectional controlling the Receiver through RS232 Protocol.

3. Example application

ELIUM IRD Receiver can be controlled from your PC. Be aware that only two wires of nine are used (RX and TX).

4. Working conditions

ELIUM IRD Receivers chars only if the following conditions are fulfilled:

- Baud: 115.200
- Parity: none
- Data Bits: 8
- Stop Bits: 1
- Flow Control: none

5. Attention:

Please mention that after switching on the unit by pushing the Power Switch, the unit is starting and during this procedure should not be disturbed. If you send anything during the starting procedure, the unit can go to Firmware update procedure. So it is recommended waiting until receive text information from application part "All units have launched properly".

6. Note:

In certain moments ELIUM Receiver sent other "#" lines too.
The syntax is: #?/text/?#

They give information about: Boot, Application Version etc. These lines should not be taken into account.

7. Commands without additional return value

Each command starts with "<" char and ends with ">". Immediately after ">" sign is received, command will be performed.

If command is not recognized (for example, if <ABC> command is sent), the following text should appear on your terminal window:

```
#COMMAND: <ABC> 3 char(s)  
#ERROR: Command not supported
```

If command is supported and was received correctly you should get:

```
#COMMAND : <ON> 2 char (s)  
#OK: Command performed
```

The line "#COMMAND:" is sent before command is performed. It only indicates that certain string of chars was received by Receiver. After that, command is performed and, if this action is finished, the line "#OK" should be sent.

In order to simplify (from programmer point of view) the reception of responses (so called confirmations) the first sign sent from Receiver is always "#". So, host should wait for "#", the next letter should indicate whether everything was all right or not (#C, #E or #C, #O)

Command	Description
<ON>	Turn on Receiver (doesn't work in normal mode)
<OFF>	Turn off Receiver (doesn't work in StandBy mode)
<REB>	Reboot Receiver. Parameters are not erased.
<SSS n>	Set RS232 Baud Rate Example values are: 9600, 19200, 38400, 115200 but user can set any speed. However, Boot Loader will not change its Baud Rate which is 115.200. Only application may work with the new speed. Baud Rate will be changed immediately. No #OK: confirmation is given but if an error occur, #ERROR: line is sent. Baud Rate value can be written to internal Receiver memory <SAV> or <OFF>. After that, application will always start with the new Baud Rate.
<TTT>	Turn to TV Mode (doesn't work in TV Mode)
<TTR>	Turn to Radio Mode (doesn't work in Radio Mode)
<PRT n>	Change TV Channel (n is a number)
<PRR n>	Change Radio Channel (n is a number)
<PRT u>	Switch TV Channel up (current Channel - 1)
<PRT d>	Switch TV Channel down (current Channel + 1)
<PRR u>	Switch Radio Channel up (current Channel - 1)
<PRR d>	Switch Radio Channel down (current Channel + 1)
<VOL n>	Change Volume (n=0 .. 100)