Encodix series

Access Module

Encodix - Edit Demo

Dafocus
http://www.dafocus.com/

1 Introduction

This document explains how to run the Encodix demonstrative project. Please, refer to Encodix manual for more details about Encodix.

1.1 What is Encodix?

Encodix is a mean to describe telecommunication message formats. Encodix provides a formal language which resembles informal ETSI documents; this allows a very natural way of describing messages formats. These descriptions are written in a file named *message description file*.
Therefore, Encodix **is not a message library**; for example, to be able to encode/decode messages described in ETSI TS 24.008 it is necessary to write a *message description file* using Encodix's syntax.
These *message description files* can be written by the users themselves our bought by third parties (Dafocus has some available).

1.2 What is the Access Module

The access module is an Encodix module which generates the required code to access algorithmically the data structures containing message definitions. This program demonstrates an advanced use of this feature by creating a graphical messages editor.

1.3 Installation

Encodix is distributed as a .ZIP file. This file must be extracted inside an empty directory; **be careful to maintain the directory sub-tree**.
The Encodix tool works under Windows 32bit (NT, 2000, 95/98/ME, etc.) and Linux. The generated code works wherever an ANSI C compiler is available.
From now on, we shall call Encodix root directory <`*Encodix_root*`>.

1.4 Demo version limits

If a license is not installed, Encodix runs in demo-mode.
Demo version of Encodix allows usage of all features but it limits message description files. The tool accepts only the included demo files and it allows to change a few lines of them. In this way, users can try all the features one at the time by editing the original demo files.
Messages exploited in this demo *message description files* are *realistic* although they might not work in real environments.

2 Access demo

The "Encodix-EDIT" demo is a 04.18/04.08 message editor. The enclosed demo file (`demo.src`) supports the following messages from 3GPP TS 04.18:

- 9.1.18, IMMEDIATE ASSIGNMENT
- 9.1.20, IMMEDIATE ASSIGNMENT REJECT
- 9.1.43a, SYSTEM INFORMATION TYPE 13

Any other message can be implemented just by updating the `demo.src` source file, running Encodix and compiling again the application!

2.1 Building the demo

Messages are described in a file named <`*Encodix_root*`>`\demo-EDIT\demo.src`.

Generate the code
First step is generation of code, done by running Encodix. Encodix can be executed through a batch file:

- open a command prompt;
- change current directory to <*Encodix_root*>`\demo-EDIT`;
- run `regen.bat`

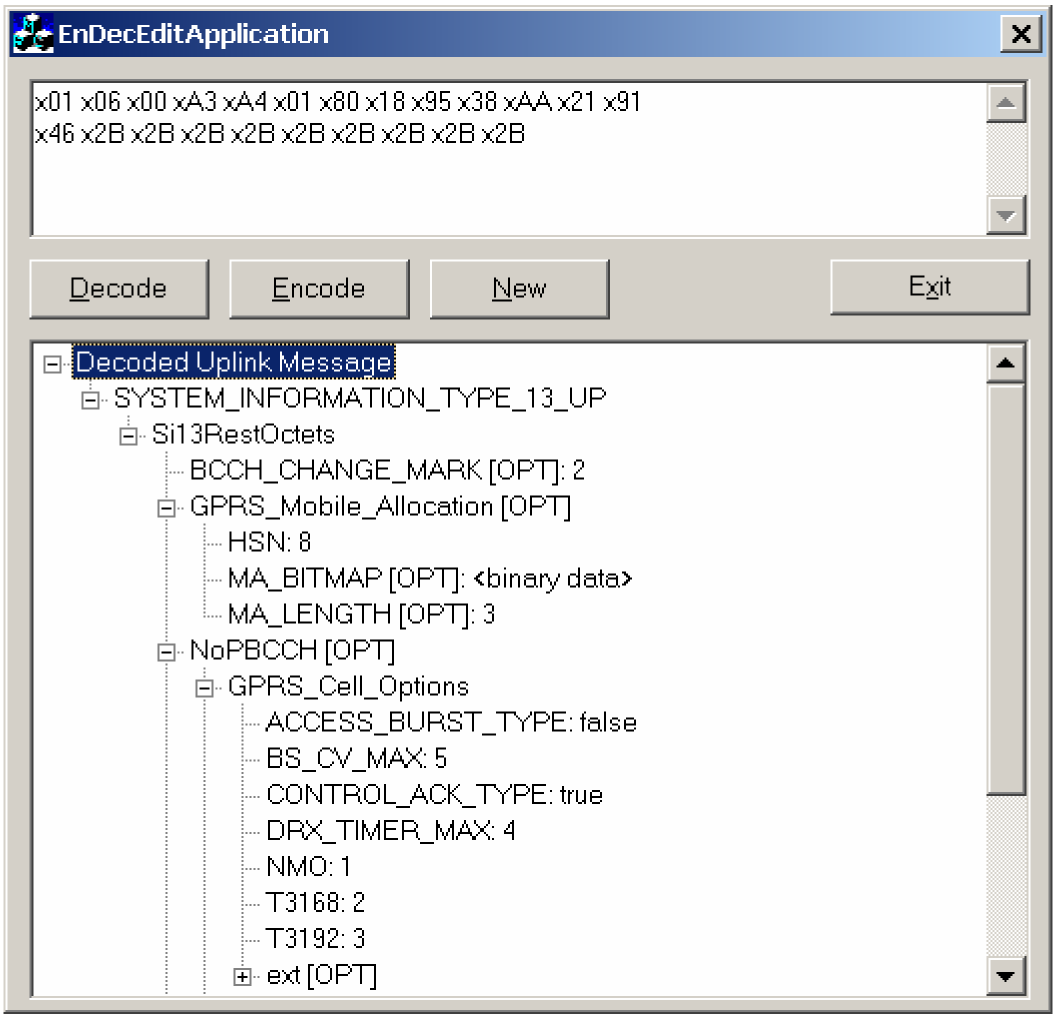
Compile the C source files
We provide with a Microsoft Visual C++ 6.0 project file under <*Encodix_root*>`\demo-EDIT\vc`.
This must be compiled in order to produce the application.

2.2 Running the application

The application should be under:

<*Encodix_root*>`\demo-EDIT\vc\EncodixEditApplication\Debug\`

its name should be `EncodixEditApplication.exe`. Once executed, the following window appears:



```
x01 x06 x00 xA3 xA4 x01 x80 x18 x95 x38 xAA x21 x91
x46 x2B x2B x2B x2B x2B x2B x2B x2B x2B
```

```
Decoded Uplink Message
  SYSTEM_INFORMATION_TYPE_13_UP
    Si13RestOctets
      BCCH_CHANGE_MARK [OPT]: 2
      GPRS_Mobile_Allocation [OPT]
        HSN: 8
        MA_BITMAP [OPT]: <binary data>
        MA_LENGTH [OPT]: 3
      NoPBCCH [OPT]
        GPRS_Cell_Options
          ACCESS_BURST_TYPE: false
          BS_CV_MAX: 5
          CONTROL_ACK_TYPE: true
          DRX_TIMER_MAX: 4
          NMO: 1
          T3168: 2
          T3192: 3
          ext [OPT]
```

The window above contains an encoded message; the window below, instead, contains a message exploded and editable.
By clicking with the right mouse button on the tree nodes, it is possible to edit the data by changing values, activating optional structure fields, adding or removing items to sequences etcetera.
The window above accepts the message according to the "BitEncode syntax" explained below.

The DECODE button will take the encoded string above and decode it in the tree below.
The ENCODE button will take the selected entry from the tree below and it will encode it in the window above.
The NEW button will create a new entry below.

A valid encoded message (System Information Type 13) to try out the application is:

```
x01 x06 x00 xA3 xA4 x01 x80 x18 x95 x38 xAA x21 x91
x46 x2B x2B x2B x2B x2B x2B x2B x2B x2B
```

2.2.1 The BitEncode syntax

The binary edit window accepts the following syntax:

One-char terminals:

<EOF>	is the end of the input string.
<SEP>	is a separator: space, tab, hypen (-), comma (,), CR, LF or <EOF>.
<DEC>	is a caracter in the following range: [0-9]
<HEX>	is a caracter in the following range: [0-9A-Fa-f]
<BIN>	is a caracter in the following range: [0-1]

Comments can be inserted inside the string betweed '<' and '>'

BNF productions:

```
string ::= (dec|bin|hex|chk)*   -- Root production.
dec ::= 'd'rep<DEC>+<SEP>   -- insert into curr pos rep-bit decimal number
hex ::= 'x'rep<HEX>+<SEP>   -- insert into curr pos rep-bit hex number
bin ::= <BIN>{n}<SEP>       -- insert into curr pos rep-bit binary value
rep ::= ['{'<DEC>+'}']      -- tells num of bits to use; defaults to 8
chk ::= '|'                 -- checks that symbol is between octets
```

Example:

```
x02 x{4}3 d{4}15 101 d{5}20
```