ECALL COMMUNICATIONS TEST BENCH
– STRUCTURE AND CONTENT OF MSD, FDS AND MDS MESSAGES

1  ECALL TEST BENCH

eCall communications test bench system simulates the operation of in-vehicle terminal, service centre and PSAP (Figure 1).

Figure 1. eCall communications test bench.

1.1  eCall test service

The eCall test service is managed through a Web user interface via secured https connection.

Figure 2. eCall test service.
1.2 Communication framework

The data content and format of eCall messages is based on eSafety forum\(^1\) eCall DG recommendations for eCall Minimum Set of Data (MSD): “Recommendations for the introduction of the pan-European eCall” (published April 2006) \(^1\).

In addition to the recommended MSD format, the test bench supports also two other non-standard special-purpose formats (called FDS and MDS), which formats are based on the eCall prestudy in Finland (within AINO programme\(^2\)) and previous specifications and drafts by E-MERGE (EU project)\(^3\), Telematics Forum (GTP 1.0)\(^4\) and ETSI (OCG EMTEL)\(^5\).

1.2.1 From in-vehicle terminal to PSAP, MSD format

In-vehicle terminal sends message containing the eCall Minimum Set of Data (MSD) to PSAP.

With the eCall test service, the data transmission is accomplished via IP-network using the HTTP POST method for testing purposes.

1.2.2 From in-vehicle terminal to PSAP, MDS and FDS formats

**MDS format**

In-vehicle terminal sends message containing the minimum data set (MDS) to PSAP. The data transmission can be accomplished either using voice channel to send DTMF codes or via IP-network (e.g. GPRS) using the HTTP POST method.

The length of binary encoded MDS message is 19 bytes. Using voice channel (phone call) the message is coded as DTMF signals. One byte produces two DTMF marks, so the length of the message is 38 DTMF marks.

**FDS format**

In-vehicle terminal sends full data set (FDS) message to service centre. The message is in XML format and is transmitted over IP-network (GPRS) using the HTTP POST method. The service centre will check the validity of the structure and content of the message using XML schema [http://www.w3.org/XML/Schema].

\(^1\) http://ec.europa.eu/information_society/activities/esafety/forum/index_en.htm
\(^2\) http://www.aino.info/indexe.html
\(^3\) http://www.gstforum.org/en/subprojects/rescue/about_gst_rescue/introduction/e-merge.htm
\(^5\) http://www.emtel.etsi.org/
Service centre forwards received valid FDS messages to PSAP. Further, the service centre can provide additional information or fill missing information. (For example, the centre may include a database containing information about the vehicle.) The additional information can be sent as separate messages (FDS+) following the original FDS message. All the messages are sent using HTTP POST.
2 THE STRUCTURE AND CONTENT OF THE MESSAGES

2.1 MSD message (eSafety forum eCall DG recommendation, April 2006)

The format of the MSD message is outlined in the following table:

<table>
<thead>
<tr>
<th>Name</th>
<th>Size (bytes)</th>
<th>Type</th>
<th>Validation</th>
<th>Description</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td>1</td>
<td>Integer</td>
<td>no</td>
<td>Bit7: Automatic activation&lt;br&gt;Bit6: Manual activation&lt;br&gt;Bit5: Test call&lt;br&gt;Bit4: No confidence in position&lt;br&gt;Bit3-Bit0: Reserved</td>
<td>2.1.1</td>
</tr>
<tr>
<td>Vehicle identification</td>
<td>20</td>
<td>String</td>
<td>The number consist of 17 characters not including the letters I, O or Q.</td>
<td>VIN number according to ISO 3779</td>
<td>2.1.2</td>
</tr>
<tr>
<td>Time stamp</td>
<td>4</td>
<td>Integer</td>
<td>value &gt;= 0</td>
<td>UTC seconds</td>
<td>2.1.3</td>
</tr>
<tr>
<td>Location</td>
<td>4</td>
<td>Integer</td>
<td>-3240000000 ≤ value ≤ 324000000</td>
<td>Latitude (WGS-84) in milliarcseconds</td>
<td>2.1.4</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Integer</td>
<td>-6480000000 ≤ value ≤ 648000000</td>
<td>Longitude (WGS-84) in milliarcseconds</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Byte</td>
<td>0 ≤ value ≤ 255</td>
<td>Direction in degrees. The nearest integer of 360.0*value/255.0</td>
<td></td>
</tr>
<tr>
<td>Service provider</td>
<td>4</td>
<td>Byte[4]</td>
<td>IPV4 format or blank field</td>
<td>Service provider IP Address or blank field</td>
<td>2.1.5</td>
</tr>
<tr>
<td>Optional data</td>
<td>102</td>
<td>String</td>
<td>no</td>
<td>Further data (e.g. crash information) or blank field</td>
<td>2.1.6</td>
</tr>
<tr>
<td><strong>Total bytes:</strong></td>
<td><strong>140</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. MSD message format.

The table column "Validation" shows what rules are used by the eCall test bench to validate the values of input data. The MSD message is encoded in 140 bytes (byte₁ – byte₁₄₀) for data transmission.
The following subsections describe the encoding of the field values of the message. (Note that all the encoding rules used are not yet completely specified in the recommendation [1] and are currently specific to the eCall test bench.)

2.1.1 Control

The control field is encoded into one byte (byte1). The bits are described in Table 1. 
(bit0 = 0x1, bit1 = 0x2, …, bit7 = 0x80)

2.1.2 Vehicle identifier

The vehicle identifier is represented by a string of 17 characters according to ISO 3779. It is encoded into 20 bytes (byte2-byte21). Each byte corresponds to one character. The last three bytes (byte19-byte21) correspond to blank characters.

2.1.3 Time stamp

Time stamp is represented as the difference (in seconds) between the current time and midnight, January 1, 1970 UTC. It is encoded as a four byte integer (byte22 – byte25; most significant byte first).

2.1.4 Location and direction

The location is defined by WGS-84 format coordinates represented in milliarcseconds. The position latitude and longitude values are signed. The allowed value ranges are shown in Table 1.

The latitude is encoded into four byte integer (byte26-byte29; most significant byte first).

The longitude is encoded into four byte integer (byte30-byte33; most significant byte first).

The direction represents the direction of travel in degrees (based on last three positions). The direction value is between 0 and 360 degrees. It is encoded into one byte (byte34) by rounding the direction value into range 0 to 255 as follows: the byte value is the nearest integer of (255.0 * direction)/ 360.0).

2.1.5 Service provider

The service provider field is optional. If the value is provided, IPV4 format should be followed.
The IPV4 address (a.b.c.d; where a,b,c and d is between 0 and 255) is encoded into four bytes (byte\textsubscript{35}-byte\textsubscript{38}) as follows. The encoded result is in network byte order: the highest order byte of the address is in the position of byte\textsubscript{35}.

If the service provider address is not provided, the four bytes corresponds to blank characters (i.e., the bytes has the numeric value 32).

2.1.6 Optional data

The optional data field is optional. The field is encoded as 102 bytes (byte\textsubscript{39}-byte\textsubscript{140}).

The input data string is encoded as follows. Each character of the string is converted into one byte (starting from the byte\textsubscript{39}). The remaining bytes are assigned the value 32 (blank character).
2.2 FDS message (Finnish proposal, June 2005)

FDS message includes the following information:

<table>
<thead>
<tr>
<th>Content</th>
<th>XML description</th>
<th>Required</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>header/flags element</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>private attribute (true or false)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>test attribute (true or false)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Message type</td>
<td>header/type element</td>
<td>X</td>
<td>2.2.1</td>
</tr>
<tr>
<td>Version</td>
<td>header/version element</td>
<td>X</td>
<td>2.2.1</td>
</tr>
<tr>
<td>Message control</td>
<td>header/control element</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>buffered attribute (true or false)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>response attribute (true or false)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Privilege level</td>
<td>header/privilege element</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vehicle type</td>
<td>ivs/vehicle/type element</td>
<td>X</td>
<td>2.2.2</td>
</tr>
<tr>
<td>Carco</td>
<td>ivs/vehicle/cargo element</td>
<td>X</td>
<td>2.2.2</td>
</tr>
<tr>
<td>Vehicle manufacturer</td>
<td>ivs/vehicle/manufacturer element</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vehicle manufacturing year</td>
<td>ivs/vehicle/model_year element</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vehicle identification number</td>
<td>ivs/vehicle/vin element</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vehicle license number</td>
<td>ivs/vehicle/license element</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vehicle colour</td>
<td>ivs/vehicle/colour element</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vehicle model</td>
<td>ivs/vehicle/model element</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Terminal MSID code</td>
<td>ivs/terminal/msid element</td>
<td>X</td>
<td>2.2.2</td>
</tr>
<tr>
<td></td>
<td>(MSISDN</td>
<td>IMEI</td>
<td>IMSI) element</td>
</tr>
<tr>
<td>Terminal manufacturer</td>
<td>ivs/terminal/manufacturer element</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Terminal HW version</td>
<td>ivs/terminal/hardware element</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Terminal SW version</td>
<td>ivs/terminal/software element</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Terminal serial number</td>
<td>ivs/terminal/serial element</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service provider IP address</td>
<td>ivs/service_provider/ip_address element</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Palveluntarjoajan puh. no.</td>
<td>ivs/service_provider/phone element</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service provider country</td>
<td>ivs/service_provider/country element</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timestamp</td>
<td>setting/time element</td>
<td>X</td>
<td>2.2.3</td>
</tr>
<tr>
<td>Current location</td>
<td>setting/current_location element</td>
<td>X</td>
<td>2.2.3</td>
</tr>
<tr>
<td>Driving direction</td>
<td>setting/direction element</td>
<td>X</td>
<td>2.2.3</td>
</tr>
<tr>
<td>Velocity</td>
<td>setting/velocity element</td>
<td>X</td>
<td>2.2.3</td>
</tr>
<tr>
<td>Previous location</td>
<td>setting/previous_location element</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 2. The content of FDS message.

In the table, the required elements of the message in XML format are shown in italics and marked with (X) in the last column. The default values of attributes are shown in italics.

The structure and semantics of the message is described by an XML schema that is located in the following public address:

http://www.ecall.fi/schemas/fds_schema.xsd

The generic XML structure of FDS message is as follows:

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<fds>
  <header>...</header>  (cf. Section 2.1.1)
  <ivs>...</ivs>  (Section 2.1.2)
  <setting>...</setting>  (Section 2.1.3)
  <incident>...</incident>  (Section 2.1.4)
  <info>Additional information</info>
</fds>
```

An example of FDS message:

http://www.ecall.fi/examples/fds_example.xml

2.2.1 FDS header

The type of the message is "FDS, which is coded in XML format ("type" element) as the number "11"."
The current message version is "1.0", which is coded in XML representation as the number "0".

```xml
<header>
  <type>11</type>
  <version>0</version>
</header>
```

### 2.2.2 Vehicle and in-vehicle terminal information

Vehicle and in-vehicle terminal information are organized as subelements of the element "ivs".

Vehicle type is coded as a number between 0-7. (reserved)

Cargo is coded as a number between 0-255 (reserved)

Message identifier (MSID) is MSISDN, IMEI or IMSI (at least one should be given in the message). The values are represented as follows [3GPP TS 23.003 V6.5.0 (2004-12)]:

**IMEI (International Mobile station Equipment Identities)**

Representation: 15 digits (always)

- TAC (Type Allocation Code): 8 digits
- + SNR (Serial Number): 6 digits
- + 1 digit

**MSISDN (Mobile Station International ISDN Number)**

Representation: in total at most 15 digits (default)

- CC (Country Code):
  - + NDC (National Destination Code):
  - + SN (Subscriber Number):

**IMSI (International Mobile Subscriber Identity)**

Representation: in total at most 15 digits
MCC (Mobile Country Code): 3 digits
+ MNC (Mobile Network Code): 2-3 digits
+ MSIN (Mobile Subscriber Identification Number)

An example:

```xml
<ivs>
  <vehicle>
    <type>11</type>
    <cargo>0</cargo>
  </vehicle>
  <terminal>
    <msid>
      <msisdn>3580123456789</msisdn>
      <imei>012345768901234</imei>
    </msid>
  </terminal>
</ivs>
```

2.2.3 Timestamp and in-vehicle movement/location information

Timestamp of current location is represented as follows [LIF MLP 3.0.0]:
year, month, day, hours, minutes, seconds

The attribute "utc_off" of the time element is optional. Its default value is utc_off=0.

All coordinates are represented in WGS-84 decimal format.

```xml
<setting>
  <time utc_off="+0200">20050613010423</time>
  <current_location>
    <coord>
      <latitude>60.123456N</latitude>
      <longitude>24.9876543E</longitude>
    </coord>
  </current_location>
  <direction>130</direction>
  <velocity>178.3</velocity>
</setting>
```
2.2.4 Incident information

Accident information include: message source and recognition, the number of passengers and other additional information. These are organised as subelements of the element "incident".

The values of message source ("source" element) belong to the following set: "manual", "rolled over", "airbag", "crash" and "moved"

```
<incident>
  <source>
    <item>rolled over</item>
    <item>airbag</item>
  </source>
  <passenger>4</passenger>
</incident>
```

2.3 MDS message (Finnish proposal, June 2005)

MDS message contains only the required data.

The message is encoded tightly to 19 bytes for sending in DTMF format within phone call.

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Content</th>
<th>Description</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Header</td>
<td>message type (5 bits) + version (3 bits)</td>
<td>Section 2.3.1</td>
</tr>
<tr>
<td>2</td>
<td>Status</td>
<td>source (5 bits) + vehicle type (3 bits)</td>
<td>2.3.2</td>
</tr>
<tr>
<td>3</td>
<td>Cargo</td>
<td>cargo type</td>
<td>2.3.2</td>
</tr>
<tr>
<td>4-10</td>
<td>Identifier</td>
<td>MSID (IMEI, IMSI or MSISDN)</td>
<td>2.3.3</td>
</tr>
<tr>
<td>11-13</td>
<td>Latitudi</td>
<td>WGS84 in degrees (signed -90 90)</td>
<td>2.3.4</td>
</tr>
<tr>
<td>14-16</td>
<td>Longitudi</td>
<td>WGS84 in degrees (signed, -180 180)</td>
<td>2.3.4</td>
</tr>
<tr>
<td>17</td>
<td>Velocity</td>
<td>km/h (0-254 and 255 when &gt;= 255) (integer)</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Direction</td>
<td>degrees * 255 / 360 (rounded to nearest integer)</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Checksum</td>
<td>CRC-8</td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Encoding of MDS message into 19 bytes.

Using DTMF codes it is possible to send only numbers 0-9, letters A-D, and marks # and *. The conversion of binary data to DTMF is performed by translating the 19 bytes into hexadecimal numbers and replacing E and F with # and *, respectively.

(For example, the byte queue "243 14 6" is coded as hexadecimal numbers as follows:
"F3 0E 06" and after the replacement the resulting DTMF sequence "*30#06".)
2.3.1 MDS header

The type of the message is "MDS" which is coded as 5-bit binary number "01011".

The current version of the message is "1.0" which is coded as 3-bit binary number "000".

Thus, the header in binary format is: 01011000.

2.3.2 Status and cargo

Message source is represented by five bits that is coded as follows:

<table>
<thead>
<tr>
<th>Source</th>
<th>Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;manual&quot;</td>
<td>1</td>
</tr>
<tr>
<td>&quot;rolled over&quot;</td>
<td>2</td>
</tr>
<tr>
<td>&quot;airbag&quot;</td>
<td>3</td>
</tr>
<tr>
<td>&quot;crash&quot;</td>
<td>4</td>
</tr>
<tr>
<td>&quot;moved&quot;</td>
<td>5</td>
</tr>
</tbody>
</table>

Vehicle type is coded as 3 bits, a number between 0-7. (reserved)

Cargo is coded as 1 byte, a number between 0-255. (reserved)

2.3.3 Coding of the MSID field of the message

The MSID field contains one of the following: IMSI, MSISDN or IMEI. Their values are represented as shown in Section 2.1.2.

Coding of the MSID identifier

The MSID identifier (number) will be translated into 7-byte binary number:

<table>
<thead>
<tr>
<th>Type</th>
<th>code</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMEI</td>
<td>1</td>
</tr>
<tr>
<td>MSISDN</td>
<td>2</td>
</tr>
<tr>
<td>IMSI</td>
<td>3</td>
</tr>
</tbody>
</table>

Type code is included into the first byte (b0) using 5 bits (bit3-bit7).
Example:
Input: IMEI identifier "001234567890123".
=> in binary format:
00000000 00000001 00011111 01110001 11111011 00000000 11001011
where bytes b0 b1 b2 b3 b4 b5 and b6.

Then, MSID type is coded to the result (IMEI => 1):
=> **00001**000 00000001 00011111 01110001 11111011 00000000 11001011

2.3.4 Coding of Latitude and longitude values

Input:

- Latitude in WGS-84 decimal format (signed [-90, 90])
- Longitude in WGS-84 decimal format (signed [-180, 180]).

The values are coded as 24-bit binary numbers as follows:

Latitude: \((\text{latitude}+90)\times(2^{24}-1)/180\) rounded to nearest integer converted to binary format.

Longitude: \((\text{longitude}+180)\times(2^{24}-1)/360\) rounded to nearest integer converted to binary format.

Example:
**latitude** = 60.123456

\((60.123456+90)\times(2^{24}-1)/180 = 13992519\) (rounded to nearest integer)
=> 11010101 10000010 01000111
where bytes b0 b1 and b2.

**longitude** = -24.123456

\((-24.123456+180)\times(2^{24}-1)/360 = 1124234\)
=> 00010001 00100111 10001010
References

1. eSafety Forum eCall DG: “Recommendations for the introduction of the pan-European eCall” (published April 2006).