



# ASN1 for eCall In-Band Modem Library

## Application Note



**SIERRA**  
WIRELESS®

4114948  
3.1  
December 14, 2015

## Important Notice

Due to the nature of wireless communications, transmission and reception of data can never be guaranteed. Data may be delayed, corrupted (i.e., have errors) or be totally lost. Although significant delays or losses of data are rare when wireless devices such as the Sierra Wireless modem are used in a normal manner with a well-constructed network, the Sierra Wireless modem should not be used in situations where failure to transmit or receive data could result in damage of any kind to the user or any other party, including but not limited to personal injury, death, or loss of property. Sierra Wireless accepts no responsibility for damages of any kind resulting from delays or errors in data transmitted or received using the Sierra Wireless modem, or for failure of the Sierra Wireless modem to transmit or receive such data.

## Safety and Hazards

Do not operate the Sierra Wireless modem in areas where cellular modems are not advised without proper device certifications. These areas include environments where cellular radio can interfere such as explosive atmospheres, medical equipment, or any other equipment which may be susceptible to any form of radio interference. The Sierra Wireless modem can transmit signals that could interfere with this equipment. Do not operate the Sierra Wireless modem in any aircraft, whether the aircraft is on the ground or in flight. In aircraft, the Sierra Wireless modem **MUST BE POWERED OFF**. When operating, the Sierra Wireless modem can transmit signals that could interfere with various onboard systems.

---

*Note: Some airlines may permit the use of cellular phones while the aircraft is on the ground and the door is open. Sierra Wireless modems may be used at this time.*

---

The driver or operator of any vehicle should not operate the Sierra Wireless modem while in control of a vehicle. Doing so will detract from the driver or operator's control and operation of that vehicle. In some states and provinces, operating such communications devices while in control of a vehicle is an offence.

## Limitations of Liability

This manual is provided "as is". Sierra Wireless makes no warranties of any kind, either expressed or implied, including any implied warranties of merchantability, fitness for a particular purpose, or noninfringement. The recipient of the manual shall endorse all risks arising from its use.

The information in this manual is subject to change without notice and does not represent a commitment on the part of Sierra Wireless. SIERRA WIRELESS AND ITS AFFILIATES SPECIFICALLY DISCLAIM LIABILITY FOR ANY AND ALL DIRECT, INDIRECT, SPECIAL, GENERAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES INCLUDING, BUT NOT LIMITED TO, LOSS OF PROFITS OR REVENUE OR ANTICIPATED PROFITS OR REVENUE ARISING OUT OF THE USE OR INABILITY TO USE ANY SIERRA WIRELESS PRODUCT, EVEN IF SIERRA WIRELESS AND/OR ITS AFFILIATES HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR THEY ARE FORESEEABLE OR FOR CLAIMS BY ANY THIRD PARTY.

Notwithstanding the foregoing, in no event shall Sierra Wireless and/or its affiliates aggregate liability arising under or in connection with the Sierra Wireless product, regardless of the number of events, occurrences, or claims giving rise to liability, be in excess of the price paid by the purchaser for the Sierra Wireless product.

Customer understands that Sierra Wireless is not providing cellular or GPS (including A-GPS) services. These services are provided by a third party and should be purchased directly by the Customer.

**SPECIFIC DISCLAIMERS OF LIABILITY:** CUSTOMER RECOGNIZES AND ACKNOWLEDGES SIERRA WIRELESS IS NOT RESPONSIBLE FOR AND SHALL NOT BE HELD LIABLE FOR ANY DEFECT OR DEFICIENCY OF ANY KIND OF CELLULAR OR GPS (INCLUDING A-GPS) SERVICES.

## Patents

This product may contain technology developed by or for Sierra Wireless Inc.

This product includes technology licensed from QUALCOMM®.

This product is manufactured or sold by Sierra Wireless Inc. or its affiliates under one or more patents licensed from InterDigital Group and MMP Portfolio Licensing.

## Copyright

© 2015 Sierra Wireless. All rights reserved.

## Trademarks

Sierra Wireless®, AirPrime®, AirLink®, AirVantage®, WISMO®, ALEOS® and the Sierra Wireless and Open AT logos are registered trademarks of Sierra Wireless, Inc. or one of its subsidiaries.

Watcher® is a registered trademark of Netgear, Inc., used under license.

Windows® and Windows Vista® are registered trademarks of Microsoft Corporation.

Macintosh® and Mac OS X® are registered trademarks of Apple Inc., registered in the U.S. and other countries.

QUALCOMM® is a registered trademark of QUALCOMM Incorporated. Used under license.

Other trademarks are the property of their respective owners.

## Contact Information

Sales Desk:	Phone:	1-604-232-1488
	Hours:	8:00 AM to 5:00 PM Pacific Time
	Contact:	<a href="http://www.sierrawireless.com/sales">http://www.sierrawireless.com/sales</a>
Post:	Sierra Wireless 13811 Wireless Way Richmond, BC Canada V6V 3A4	
Technical Support:	<a href="mailto:support@sierrawireless.com">support@sierrawireless.com</a>	
RMA Support:	<a href="mailto:repairs@sierrawireless.com">repairs@sierrawireless.com</a>	
Fax:	1-604-231-1109	
Web:	<a href="http://www.sierrawireless.com/">http://www.sierrawireless.com/</a>	

Consult our website for up-to-date product descriptions, documentation, application notes, firmware upgrades, troubleshooting tips, and press releases: [www.sierrawireless.com](http://www.sierrawireless.com)

# Document History

Version	Date	Updates
1.0	October 20, 2011	Creation
2.0	February 14, 2014	Updated legal boilerplate content. Correction to subsection 2.12, size corrected as $\geq 140$ ; was formerly $\leq$ . Corrected 2.12.2, <code>EncodeMSDMessage()</code> returns the number of bits not bytes. Clarified in 3.2 that <code>vehiclePropulsionStorageType</code> must be entered in hexadecimal. Added chapter 3 with <code>asn1_EraGlonass_EncodeOptionalData()</code> functionality.
2.1	July 14, 2014	Updated legal boilerplate content. Updated definition in Section 2.8.1 and field definitions in Section 2.8.2. Updated information regarding MSD V2 compliance including the Read Command in Section 4.1.2 and the <code>&lt;vehiclePropulsionStorageType&gt;</code> parameter definition in Section 4.2.
3.0	November 9, 2014	Updated <code>eraGlonass_AdditionalData_t</code> to match the new scheme of September 2014. Clarified value ranges.
3.1	December 14, 2015	Updated: <ul style="list-style-type: none"><li>• Legal boilerplate content.</li><li>• Related Documents section.</li></ul>

# Contents

<b>CONTENTS .....</b>	<b>5</b>
<b>1. INTRODUCTION .....</b>	<b>7</b>
1.1. Overview.....	7
1.2. Related Documents.....	7
1.3. Abbreviations.....	7
<b>2. ASN1 CONTROL API.....</b>	<b>8</b>
2.1. Required Header File .....	8
2.2. The eCallMessage_t type.....	8
2.2.1. Definition .....	8
2.2.2. Fields.....	8
2.3. The MSDMessage_t type.....	8
2.3.1. Definition .....	8
2.3.2. Fields.....	9
2.4. The MSDStructure_t type.....	9
2.4.1. Definition .....	9
2.4.2. Fields.....	9
2.5. The ControlType_t type.....	10
2.5.1. Definition .....	10
2.5.2. Fields.....	11
2.6. The e_VehicleType type.....	11
2.6.1. Definition .....	11
2.6.2. Value .....	11
2.7. The VIN_t type.....	12
2.7.1. Definition .....	12
2.7.2. Fields.....	12
2.8. The VehiclePropulsionStorageType_t type.....	13
2.8.1. Definition .....	13
2.8.2. Fields.....	13
2.9. The VehicleLocation_t type.....	13
2.9.1. Definition .....	13
2.9.2. Fields.....	14
2.10. The VehicleLocationDelta_t type.....	14
2.10.1. Definition .....	14
2.10.2. Fields.....	14
2.11. The OptionalData_t type .....	15
2.11.1. Definition .....	15
2.11.2. Fields.....	15
2.12. The msd_encode API.....	15
2.12.1. Prototype .....	15
2.12.2. Parameters.....	15

---

2.12.3. Returned Values .....	15
<b>3. ERA GLONASS CONTROL API EXTENSION FOR ADDITIONALDATA.....</b>	<b>17</b>
<b>3.1. Introduction.....</b>	<b>17</b>
<b>3.2. Required Header File .....</b>	<b>17</b>
<b>3.3. The eraGlonass_AdditionalData_t type .....</b>	<b>17</b>
3.3.1. ASN.1 Scheme.....	17
3.3.2. Definition .....	18
3.4. struct diagnosticResult .....	22
<b>3.4.1. Fields .....</b>	<b>22</b>
<b>3.5. struct CrashType_t.....</b>	<b>24</b>
<b>3.5.1. Fields .....</b>	<b>24</b>
<b>3.6. The asn1_EraGlonass_EncodeOptionalData() function .....</b>	<b>25</b>
3.6.1. Prototype.....	25
3.6.2. Parameters.....	25
3.6.3. Returned Values .....	25
<b>4. ASN1 AT COMMAND .....</b>	<b>26</b>
4.1. Command +ASN1MSD .....	26
4.1.1. Description .....	26
4.1.2. Syntax .....	26
4.2. Parameters and Defined Values .....	27
4.3. Examples.....	29

# 1. Introduction

## 1.1. Overview

This application note provides detailed information about the ASN1 coder used with the eCall In-Band Modem Library.

The ASN1 coder enables customers to create the binary format of MSD needed by the IVS to be sent during eCall process to PSAP.

## 1.2. Related Documents

- [1] eCall In-Band Modem Library AT Commands Interface Guide  
Reference: 4112357
- [2] eCall In-Band Modem Library Development Guide  
Reference: 4112356
- [3] Road transport and traffic telematics — ESafety — ECall minimum set of data (CEN/ FprEN 15722:2014 E)
- [4] 3GPP TS 26.267 V9.3.0 - V8.6.0
- [5] 3GPP TS 26.268 V9.4.0 - V8.6.0
- [6] 3GPP TS 26.269 V9.2.0 - V8.3.0
- [7] GOST R 54620-2011

## 1.3. Abbreviations

Abbreviation	Definition
ADL	Application Development Layer
ASN1	Abstract Syntax Notation
GSM	Global System for Mobile Communication
IVS	In-Vehicle System
mas	milliarcsecond
MSD	Minimum Set of Data
PSAP	Public Safety Answering Point

## 2. ASN1 Control API

### 2.1. Required Header File

The header file for the ASN1 coder use is `asn1_msd.h`.

### 2.2. The `eCallMessage_t` type

The `eCallMessage_t` structure corresponds to the MSD structure provided by CEN 1.1 with the following implementation.

#### 2.2.1. Definition

```
typedef struct {
    u8 id;
    MSDMessage_t msd;
} eCallMessage_t;
```

#### 2.2.2. Fields

Field	Definition	Type	Min	Max
Id	The MSD version.	u8	1	2
Msd	Structure for msd message	MSDMessage_t		

### 2.3. The `MSDMessage_t` type

#### 2.3.1. Definition

```
typedef struct {
    MSDStructure_t msdStructure;
    /* OPTIONAL */
    bool optionalDataPres;
    OptionalData_t OptionalData;
}MSDMessage_t;
```

## 2.3.2. Fields

Field	Definition	Type
msdStructure	Structure for msd parameters	MSDStructure_t
optionalDataPres	Boolean for optional data parameter presence	Bool
OptionalData	Structure for optional data	OptionalData_t

## 2.4. The MSDStructure\_t type

### 2.4.1. Definition

```
typedef struct {
    u8    messageIdentifier;
    ControlType_t    control;
    VIN_t    vehicleIdentificationNumber;
    VehiclePropulsionStorageType_t    vehiclePropulsionStorageType;
    u32    timestamp;
    VehicleLocation_t    vehicleLocation;
    u8    vehicleDirection;
    /* OPTIONAL */
    bool    recentVehicleLocationN1Pres;
    VehicleLocationDelta_t    recentVehicleLocationN1;
    /* OPTIONAL */
    bool    recentVehicleLocationN2Pres;
    VehicleLocationDelta_t    recentVehicleLocationN2;
    /* OPTIONAL */
    bool    numberOfPassengersPres;
    u8    numberOfPassengers;
} MSDStructure_t;
```

### 2.4.2. Fields

Field	Type	Definition	Min	Max	Values with special meaning
messageIdentifier	u8	Parameter for message identifier value	0	255	-
Control	ControlType_t	Structure for control parameter	-	-	-

Field	Type	Definition	Min	Max	Values with special meaning
vehicleIdentificationNumber	VIN_t	Structure for vehicle identification number parameter	-	-	-
vehiclePropulsionStorageType	VehiclePropulsionStorageType_t	Structure for vehicle propulsion Storage type parameter	-	-	-
Timestamp	u32	Parameter for timestamp value. Seconds since 1970:00:00:00	0	4294967295 = 0xFFFFFFF	0: failed to get time
vehicleLocation	VehicleLocation_t	Structure for vehicle location parameter	-	-	-
vehicleDirection	u8	Parameter for vehicle direction value. Degrees from magnetic North, 2-deg steps.	0	179	0xFF: direction is invalid or unknown
recentVehicleLocationN1Pres	bool	Boolean for recent vehicle Location N1 parameter presence	-	-	-
recentVehicleLocationN1	VehicleLocationDelta_t	Structure for recent vehicle Location N1 parameter. Not used if recentVehicleLocationN1Pres == FALSE.	-	-	-
recentVehicleLocationN2Pres	bool	Boolean for recent vehicle Location N2 parameter presence	-	-	-
recentVehicleLocationN2	VehicleLocationDelta_t	Structure for recent vehicle Location N2 parameter. Not used if recentVehicleLocationN2Pres == FALSE.	-	-	-
numberOfPassengersPres	bool	Boolean for number of passengers parameter presence	-	-	-
numberOfPassengers	u8	Parameter for number of passengers value. Not used if numberOfPassengersPres == FALSE	0	255	255: invalid or unknown

## 2.5. The ControlType\_t type

### 2.5.1. Definition

```
typedef struct {
    bool    automaticActivation;
    bool    testCall;
```

```

    bool    positionCanBeTrusted;
    e_VehicleType    vehicleType;
} ControlType_t;

```

## 2.5.2. Fields

Field	Definition	Type
automaticActivation	Boolean for automatic Activation parameter	Bool
testCall	Boolean for test Call parameter	bool
positionCanBeTrusted	Boolean for position Can Be Trusted parameter	Bool
vehicleType	enum for vehicleType parameter	e_VehicleType

## 2.6. The e\_VehicleType type

### 2.6.1. Definition

```

typedef enum {
    VehicleType_passengerVehicleClassM1 = 1,
    VehicleType_busesAndCoachesClassM2 ,
    VehicleType_busesAndCoachesClassM3 ,
    VehicleType_lightCommercialVehiclesClassN1 ,
    VehicleType_heavyDutyVehiclesClassN2 ,
    VehicleType_heavyDutyVehiclesClassN3 ,
    VehicleType_motorcyclesClassL1e ,
    VehicleType_motorcyclesClassL2e ,
    VehicleType_motorcyclesClassL3e ,
    VehicleType_motorcyclesClassL4e ,
    VehicleType_motorcyclesClassL5e ,
    VehicleType_motorcyclesClassL6e ,
    VehicleType_motorcyclesClassL7e
} e_VehicleType;

```

### 2.6.2. Value

Value	Definition
VehicleType_passengerVehicleClassM1	Enum value for passenger vehicle Class M1 vehicle type
VehicleType_busesAndCoachesClassM2	Enum value for buses and coaches Class M2 vehicle type

Value	Definition
VehicleType_busesAndCoachesClassM3	Enum value for buses and coaches Class M3 vehicle type
VehicleType_lightCommercialVehiclesClassN1	Enum value for light commercial Class N1 vehicle type
VehicleType_heavyDutyVehiclesClassN2	Enum value for heavy and duty Class N2 vehicle type
VehicleType_heavyDutyVehiclesClassN3	Enum value for heavy and duty Class N3 vehicle type
VehicleType_motorcyclesClassL1e	Enum value for motorcycles Class L1e vehicle type
VehicleType_motorcyclesClassL2e	Enum value for motorcycles Class L2e vehicle type
VehicleType_motorcyclesClassL3e	Enum value for motorcycles Class L3e vehicle type
VehicleType_motorcyclesClassL4e	Enum value for motorcycles Class L4e vehicle type
VehicleType_motorcyclesClassL5e	Enum value for motorcycles Class L5e vehicle type
VehicleType_motorcyclesClassL6e	Enum value for motorcycles Class L6e vehicle type
VehicleType_motorcyclesClassL7e	Enum value for motorcycles Class L7e vehicle type

## 2.7. The VIN\_t type

### 2.7.1. Definition

The MSD is including fields to set the Vehicle Identification Number (VIN). The Open AT Application can set these fields using the structure below:

```
typedef struct {
    char isowmi[3];
    char isovds[6];
    char isovisModelyear[1];
    char isovisSeqPlant[7];
} VIN_t;
```

### 2.7.2. Fields

Please note that the above mentioned fields accepts only a limited set of characters. Here are the possible values as stated by ISO 3779 standard:

```
-- isowmi: World Manufacturer Index (WMI)
-- isovds: Vehicle Type Descriptor (VDS)
-- Vehicle Identifier Section (VIS) consisting of
-- isovisModelyear: Modelyear from Vehicle Identifier Section (VIS)
-- isovisSeqPlant: Plant code + sequential number
-- from Vehicle Identifier Section (VIS)
VIN ::= SEQUENCE {
    isowmi PrintableString (SIZE(3))
    (FROM ("A".."H"|"J".."N"|"P"|"R".."Z"|"0".."9")),
    isovds PrintableString (SIZE(6))
    (FROM ("A".."H"|"J".."N"|"P"|"R".."Z"|"0".."9")),
    isovisModelyear PrintableString (SIZE(1))
    (FROM ("A".."H"|"J".."N"|"P"|"R".."Z"|"0".."9")),
    isovisSeqPlant PrintableString (SIZE(7))
    (FROM ("A".."H"|"J".."N"|"P"|"R".."Z"|"0".."9"))
}
```

Field	Definition	Type
Isowmi	parameter for World Manufacturer Index	char[3]
isovds	parameter for Vehicle Type Descriptor	char[6]
isovisModelyear	parameter for Model Year	char[1]
isovisSeqPlant	parameter for Plant code + sequential number	char[7]

## 2.8. The VehiclePropulsionStorageType\_t type

### 2.8.1. Definition

```
typedef struct {
    bool gasolineTankPresent    /* DEFAULT FALSE */;
    bool dieselTankPresent    /* DEFAULT FALSE */;
    bool compressedNaturalGas  /* DEFAULT FALSE */;
    bool liquidPropaneGas     /* DEFAULT FALSE */;
    bool electricEnergyStorage /* DEFAULT FALSE */;
    bool hydrogenStorage      /* DEFAULT FALSE */;
    bool otherStorage /* DEFAULT FALSE */;
} VehiclePropulsionStorageType_t;
```

### 2.8.2. Fields

Field	Definition	Type
gasolineTankPresent	Boolean for gasoline tank presence	bool
dieselTankPresent	Boolean for diesel tank presence	bool
compressedNaturalGas	Boolean for compressed natural gas presence	bool
liquidPropaneGas	Boolean for liquid propane presence	bool
electricEnergyStorage	Boolean for electric energy storage presence	bool
hydrogenStorage	Boolean for hydrogen storage presence	bool
otherStorage	Boolean for other storage presence, used only for MSD V2	bool

## 2.9. The VehicleLocation\_t type

### 2.9.1. Definition

```
typedef struct {
    s32    positionLatitude;
    s32    positionLongitude;
```

```
} VehicleLocation_t;
```

Here is a formula for converting a value from degree to mas (milliarcsecond) unit:

```
signed32bits_myVal_mas = double_myVal_deg * 3600000;
```

## 2.9.2. Fields

Field	Definition	Unit	Min	Max	Values with special meaning
s32 positionLatitude	parameter of positionLatitude value	mas	-324000000 (-90°)	+324000000 (+90°)	0x7FFFFFFF = 2147483647: latitude is invalid or unknown
s32 positionLongitude	parameter of positionLongitude value	mas	-648000000 (-180°)	+648000000 (+180°)	0x7FFFFFFF = 2147483647: longitude is invalid or unknown

## 2.10. The VehicleLocationDelta\_t type

### 2.10.1. Definition

```
typedef struct {
    s32 latitudeDelta;
    s32 longitudeDelta;
} VehicleLocationDelta_t;
```

### 2.10.2. Fields

Field	Definition	Unit	Min	Max	Values with special meaning
s32 latitudeDelta	The delta from the position reported.	100-mas steps (latitude delta)	-512 (51.2"S)	+511 (51.1"N)	-
s32 longitudeDelta	The delta from the position reported.	100-mas steps (longitude delta)	-648000000 (-180°)	+648000000 (+180°)	-

## 2.11. The OptionalData\_t type

### 2.11.1. Definition

```
typedef struct {
    u8 oidlen;
    u8 datalen;
    u8* oid;
    u8* data;
}OptionalData_t;
```

### 2.11.2. Fields

Field	Definition	Type
oidlen	Length of OID parameter	u8
datalen	Length of data parameter	u8
oid	Pointer on OID array	u8*
data	Pointer on data array	u8*

## 2.12. The msd\_encode API

The OutputData is provided as binary. The length of the OutputData data must be greater than or equal to 140 bytes.

### 2.12.1. Prototype

```
s32 EncodeMSDMessage(eCallMessage_t* MSDData, u8 * OutputData);
```

### 2.12.2. Parameters

**MSDData:**

In.: The structure of eCall message with parameters.

**OutputData:**

Out.: The binary MSD coded with ASN1 protocol.

### 2.12.3. Returned Values

This API returns:

- Positive value on success. This value corresponding with the number of bits used to code the eCall message.
- On error, negative error codes as defined in the table below.

Error Code	Description
-1	ASN1 coding error



## 3. ERA GLONASS Control API Extension for AdditionalData

### 3.1. Introduction

This chapter describes the ERA Glonass specific parts of the MSDs AdditionalData. The data describes the seriousness of the road accident.

This function will act as a utility function and the results will be suitable to be passed to the OptionalData\_t as mentioned above and passed to EncodeMSDMessage().

Please refer to the documentation GOST 1.1 for more details about all ASN.1 references etc.

### 3.2. Required Header File

These functions also use the header file asn1\_msd.h.

### 3.3. The eraGlonass\_AdditionalData\_t type

The eraGlonassAdditionalData\_t structure contains the data corresponding to the MSD structure provided updated by September 2014.

#### 3.3.1. ASN.1 Scheme

This is the ASN.1 Scheme describing the encoding.

```
DiagnosticResult ::= SEQUENCE {
    micConnectionFailure BOOLEAN OPTIONAL,
    micFailure BOOLEAN OPTIONAL,
    rightSpeakerFailure BOOLEAN OPTIONAL,
    leftSpeakerFailure BOOLEAN OPTIONAL,
    speakersFailure BOOLEAN OPTIONAL,
    ignitionLineFailure BOOLEAN OPTIONAL,
    uimFailure BOOLEAN OPTIONAL,
    statusIndicatorFailure BOOLEAN OPTIONAL,
    batteryFailure BOOLEAN OPTIONAL,
    batteryVoltageLow BOOLEAN OPTIONAL,
    crashSensorFailure BOOLEAN OPTIONAL,
    firmwareImageCorruption BOOLEAN OPTIONAL,
    commModuleInterfaceFailure BOOLEAN OPTIONAL,
    gnssReceiverFailure BOOLEAN OPTIONAL,
    raimProblem BOOLEAN OPTIONAL,
```

```
gnssAntennaFailure BOOLEAN OPTIONAL,  
commModuleFailure BOOLEAN OPTIONAL,  
eventsMemoryOverflow BOOLEAN OPTIONAL,  
crashProfileMemoryOverflow BOOLEAN OPTIONAL,  
otherCriticalFailures BOOLEAN OPTIONAL,  
otherNotCriticalFailures BOOLEAN OPTIONAL  
}  
  
CrashInfo ::= SEQUENCE {  
    crashFront BOOLEAN OPTIONAL,  
    crashLeft BOOLEAN OPTIONAL,  
    crashRight BOOLEAN OPTIONAL,  
    crashRear BOOLEAN OPTIONAL,  
    crashRollover BOOLEAN OPTIONAL,  
    crashSide BOOLEAN OPTIONAL,  
    crashFrontOrSide BOOLEAN OPTIONAL,  
    crashAnotherType BOOLEAN OPTIONAL  
}
```

### 3.3.2. Definition

```
typedef struct {  
    bool present_crashSeverityASI15;  
    bool present_diagnosticResult;  
    bool present_crashInfo;  
    u32 crashSeverityASI15;  
    struct  
    {  
        bool present_micConnectionFailure;  
        bool micConnectionFailure;  
  
        bool present_micFailure;  
        bool micFailure;  
  
        bool present_rightSpeakerFailure;  
        bool rightSpeakerFailure;  
  
        bool present_leftSpeakerFailure;
```

```
bool leftSpeakerFailure;

bool present_speakersFailure;
bool speakersFailure;

bool present_ignitionLineFailure;
bool ignitionLineFailure;

bool present_uimFailure;
bool uimFailure;

bool present_statusIndicatorFailure;
bool statusIndicatorFailure;

bool present_batteryFailure;
bool batteryFailure;

bool present_batteryVoltageLow;
bool batteryVoltageLow;

bool present_crashSensorFailure;
bool crashSensorFailure;

bool present_firmwareImageCorruption;
bool firmwareImageCorruption;

bool present_commModuleInterfaceFailure;
bool commModuleInterfaceFailure;

bool present_gnssReceiverFailure;
bool gnssReceiverFailure;

bool present_raimProblem;
bool raimProblem;

bool present_gnssAntennaFailure;
bool gnssAntennaFailure;

bool present_commModuleFailure;
```

```
bool commModuleFailure;

bool present_eventsMemoryOverflow;
bool eventsMemoryOverflow;

bool present_crashProfileMemoryOverflow;
bool crashProfileMemoryOverflow;

bool present_otherCriticalFailires;
bool otherCriticalFailires;

bool present_otherNotCriticalFailures;
bool otherNotCriticalFailures;
}diagnosticResult;

bool crashTypePres;
struct
{
    bool present_crashFront;
    bool crashFront;

    bool present_crashLeft;
    bool crashLeft;

    bool present_crashRight;
    bool crashRight;

    bool present_crashRear;
    bool crashRear;

    bool present_crashRollover;
    bool crashRollover;

    bool present_crashSide;
    bool crashSide;

    bool present_crashFrontOrSide;
    bool crashFrontOrSide;
```

```

bool present_crashAnotherType;
bool crashAnotherType;
}crashType;
}eraGlonass_AdditionalData_t;
    
```

Field	Definition	Type
present_crashSeverityASI15	Defines if the optional element <i>crashSeverityASI15</i> should be taken into account.	bool
present_diagnosticResult	Defines if the optional element <i>diagnosticResult</i> should be taken into account.	bool
present_crashInfo	Defines if the optional element <i>crashInfo</i> should be taken into account.	bool
crashSeverityASI15	Parameter for the probability of danger to life and health of people in the vehicle. Value between 0..2047	u32
diagnosticResult	An internal structure holding all the individual diagnostic results.	Struct
crashType	An internal structure holding all the crash type results. Corresponding to the optional ASN.1 field "CrashType".	Struct

## 3.4. struct diagnosticResult

Internal structure of eraGlonass\_AdditionalData\_t.

### 3.4.1. Fields

Field	Definition	Type
present_micConnectionFailure	Indicates if the corresponding optional variable should be taken into account or not.	bool
micConnectionFailure	incorrect connection of microphone	bool
present_micFailure	Indicates if the corresponding optional variable should be taken into account or not.	bool
micFailure	operational unavailability of microphone	bool
present_rightSpeakerFailure	Indicates if the corresponding optional variable should be taken into account or not.	bool
rightSpeakerFailure	fault in right speaker	bool
present_leftSpeakerFailure	Indicates if the corresponding optional variable should be taken into account or not.	bool
leftSpeakerFailure	fault in left speaker	bool
present_speakersFailure	Indicates if the corresponding optional variable should be taken into account or not.	bool
speakersFailure	fault in speakers	bool
present_ignitionLineFailure	Indicates if the corresponding optional variable should be taken into account or not.	bool
ignitionLineFailure	fault in definition of ignition line status definition;	bool
present_uimFailure	Indicates if the corresponding optional variable should be taken into account or not.	bool
uimFailure	fault in user interface	bool
present_statusIndicatorFailure	Indicates if the corresponding optional variable should be taken into account or not.	bool
statusIndicatorFailure	fault in status indicator	bool
present_batteryFailure	Indicates if the corresponding optional variable should be taken into account or not.	bool
batteryFailure	fault in spare battery	bool
present_batteryVoltageLow	Indicates if the corresponding optional variable should be taken into account or not.	bool
batteryVoltageLow	charge in spare battery below allowable level	bool

Field	Definition	Type
present_crashSensorFailure	Indicates if the corresponding optional variable should be taken into account or not.	bool
crashSensorFailure	fault in automatic road accident identification sensor	bool
present_firmwareImageCorruption	Indicates if the corresponding optional variable should be taken into account or not.	bool
firmwareImageCorruption	violation of integrity of software image;	bool
present_commModuleInterfaceFailure	Indicates if the corresponding optional variable should be taken into account or not.	bool
commModuleInterfaceFailure	no operational availability of interface with GSM and UMTS communication module	bool
present_gnssReceiverFailure	Indicates if the corresponding optional variable should be taken into account or not.	bool
gnssReceiverFailure	no operational availability of GNSS receiver.	bool
present_raimProblem	Indicates if the corresponding optional variable should be taken into account or not.	bool
raimProblem	absence of integrity (reliability) of the specified GNSS receiver navigational and time parameters (RAIM function)	bool
present_gnssAntennaFailure	Indicates if the corresponding optional variable should be taken into account or not.	bool
gnssAntennaFailure	no operational availability (incorrect wiring) of GNSS internal antenna.	bool
present_commModuleFailure	Indicates if the corresponding optional variable should be taken into account or not.	bool
commModuleFailure	no operational availability (incorrect connection) of external GSM and UMTS antennas	bool
present_eventsMemoryOverflow	Indicates if the corresponding optional variable should be taken into account or not.	bool
eventsMemoryOverflow	overfilling of internal memory for events	bool
present_crashProfileMemoryOverflow		bool
crashProfileMemoryOverflow	overfilling of memory for recording acceleration profile	bool
present_otherCriticalFailires	Indicates if the corresponding optional variable should be taken into account or not. Spelling matches spelling in ASN.1 specification.	bool
otherCriticalFailires	other critical errors	bool

Field	Definition	Type
present_otherNotCriticalFailures;	Indicates if the corresponding optional variable should be taken into account or not.	bool
otherNotCriticalFailures	other non-critical errors	bool

## 3.5. struct CrashType\_t

Internal structure to the eraGlonass\_AdditionalData\_t .

### 3.5.1. Fields

Field	Definition	Type
present_crashFront	Indicates if the corresponding optional variable should be taken into account or not.	bool
crashFront	front collision	bool
present_crashLeft	Indicates if the corresponding optional variable should be taken into account or not.	bool
crashLeft	Left collision.	bool
present_crashRight	Indicates if the corresponding optional variable should be taken into account or not.	bool
crashRight	Right collision	bool
present_crashRear	Indicates if the corresponding optional variable should be taken into account or not.	bool
crashRear	rear impact	bool
present_crashRollover	Indicates if the corresponding optional variable should be taken into account or not.	bool
crashRollover	rollover	bool
present_crashSide	Indicates if the corresponding optional variable should be taken into account or not.	bool
crashSide	side collision	bool
present_crashFrontOrSide	Indicates if the corresponding optional variable should be taken into account or not.	bool
crashFrontOrSide	front or side collision	bool
present_crashAnotherType	Indicates if the corresponding optional variable should be taken into account or not.	bool
crashAnotherType	other types of damage	bool

## 3.6. The `asn1_EraGlonass_EncodeOptionalData()` function

The utility function provides a way to encode the ERA GLONASS crash profile values into ASN.1 code suitable to set in the Optional Data part of the generic `MSDMessage_t` structure. Please note that the calling function provides the buffer for the `pOutData`.

### 3.6.1. Prototype

```
s32 asn1_EraGlonass_EncodeOptionalData ( eraGlonass_AdditionalData_t  
inEraAdditonalData , u8* pOutData );
```

### 3.6.2. Parameters

**inEraAdditonalData:**

In: The pointer containing the value of the ERA GLONASS crash profile values.

**pOutData:**

Out: This is a buffer allocated by the calling function. It must be at least 140 bytes big.

### 3.6.3. Returned Values

This API returns:

Positive value on success. The value is the size of `pOutData` in bits.

On error, negative error codes as defined in the table below.

Error Code	Description
-1	ASN1 coding error

## 4. ASN1 AT Command

### 4.1. Command +ASN1MSD

#### 4.1.1. Description

This AT Command is used to update MSD parameters fields and generate a binary MSD coded with ASN1 protocol.

This part is independent of the MSD used by the IVS sample of the eCall In-Band Modem Library. In order to update the MSD of IVS, the +INBMOPEN and/or +INBMSETMSD commands are to be used with the binary MSD provide with AT+ASN1MSD=2 command.

#### 4.1.2. Syntax

If <mode> = 0

*Action command*

```
AT+ASN1MSD=<mode>,<action>,<ID>,<messageIdentifier>,<Control>,<vehicleType>,<isowmi>,<isovds>,<isovisModelYear>,<isovisSeqPlant>,<vehiclePropulsionStorageType>,<timeStamp>,<vehicleLocation.PositionLatitude>,<vehicleLocation.PositionLongitude>,<vehicleDirection>
```

OK

If <mode> = 1

*Action command*

```
AT+ASN1MSD=<mode>,<action>,<recentVehicleLocationN1.latitudeDelta>,<recentVehicleLocationN1.longitudeDelta>,<recentVehicleLocationN2.latitudeDelta>,<recentVehicleLocationN2.longitudeDelta>,<numberOfPassengers>,<additionalData.oid>,<additionalData.data>
```

OK

If <mode> = 2

*Action command*

```
AT+ASN1MSD=<mode>  
+ASN1MSD=<Binary MSD>
```

OK

*Read command*

```
AT+ASN1MSD?
```

```
+ASN1MSD:
For MSD V1
<ID>,<messageIdentifier>,<automaticActivation>,<testCall>,<positionCanBeTrusted>,<vehicleType>,<isowmi>,<isovds>,<isovisModelYear>,<isovisSeqPlant>,<gasolineTankPresent>,<dieselTankPresent>,<compressedNaturalGas>,<liquidPropaneGas>,<electricEnergyStorage>,<hydrogenStorage>,<timeStamp>,<vehicleLocation.PositionLatitude>,<vehicleLocation.PositionLongitude>,<vehicleDirection>,<recentVehicleLocationN1.latitudeDelta>,<recentVehicleLocationN1.longitudeDelta>,<recentVehicleLocationN2.latitudeDelta>,<recentVehicleLocationN2.longitudeDelta>,<numberOfPassengers>,<additionalData.oid>,<additionalData.data>
For MSD V2
<ID>,<messageIdentifier>,<automaticActivation>,<testCall>,<positionCanBeTrusted>,<vehicleType>,<isowmi>,<isovds>,<isovisModelYear>,<isovisSeqPlant>,<gasolineTankPresent>,<dieselTankPresent>,<compressedNaturalGas>,<liquidPropaneGas>,<electricEnergyStorage>,<hydrogenStorage>,<otherStorage>,<timeStamp>,<vehicleLocation.PositionLatitude>,<vehicleLocation.PositionLongitude>,<vehicleDirection>,<recentVehicleLocationN1.latitudeDelta>,<recentVehicleLocationN1.longitudeDelta>,<recentVehicleLocationN2.latitudeDelta>,<recentVehicleLocationN2.longitudeDelta>,<numberOfPassengers>,<additionalData.oid>,<additionalData.data>

OK
```

```
Test command
AT+ASN1MSD=?
OK
```

## 4.2. Parameters and Defined Values

Parameter	Defined Value/Definition
<b>&lt;mode&gt;:</b>	The parameter specifies the mode
0	Mandatory parameters of MSD
1	Optional parameters of MSD
2	Encode the MSD with current parameters values
<b>&lt;action&gt; :</b>	
0	Reset the current parameters values.
1	Set the corresponding parameters with provided values.
<b>&lt;ID&gt;</b>	Integer for ID
<b>&lt;messageIdentifier&gt;</b>	Integer for messageIdentifier according 1.1
<b>&lt;control&gt;</b>	Bit mask for automaticActivation, testCall and positionCanBeTrusted MSD parameters bits set to 1 for TRUE bits set to 0 for FALSE

Parameter		Defined Value/Definition
	0	all parameters set to FALSE
	1 (bit 0)	automaticActivation parameter set to TRUE
	2 (bit 1)	testCall parameter set to TRUE
	4 (bit 2)	positionCanBeTrusted parameter set to TRUE
<b>&lt;vehicleType&gt;</b>		Integer value according 1.1
<b>&lt;isowmi&gt;</b>		ASCII string according 1.1
<b>&lt;isovds&gt;</b>		ASCII string according 1.1
<b>&lt;isovisModelYear&gt;</b>		ASCII string according 1.1
<b>&lt;isovisSeqPlant&gt;</b>		ASCII string according 1.1
<b>&lt;vehiclePropulsionStorageType&gt;</b>		Bit mask for vehicleStorageType parameter. Value entered in hexadecimal.  bits set to 1 for TRUE bits set to 0 for FALSE
	0	all values set to FALSE
	1 (bit 0)	gasolineTankPresent parameter set to TRUE
	2 (bit 1)	dieselTankPresent parameter set to TRUE
	4 (bit 2)	compressedNaturalGas parameter set to TRUE
	8 (bit 3)	liquidPropaneGas parameter set to TRUE
	10 (bit 4)	electricEnergyStorage parameter set to TRUE
	20 (bit 5)	hydrogenStorage parameter set to TRUE
	40 (bit 6)	otherStorage parameter set to TRUE, only for MSD V2
<b>&lt;timeStamp&gt;</b>		Integer for timestamp value according 1.1
<b>&lt;locationLatitude&gt;</b>		Integer for location Latitude value according 1.1
<b>&lt;locationLongitude&gt;</b>		Integer for location Longitude value according 1.1
<b>&lt;vehicleDirection&gt;</b>		Integer for vehicle Direction according 1.1
<b>&lt;recentVehicleLocation N1.latitudeDelta&gt;</b>		Integer for location Latitude value according 1.1 - OPTIONAL
<b>&lt;recentVehicleLocation N1.longitudeDelta&gt;</b>		Integer for location Longitude value according 1.1 – OPTIONAL
<b>&lt;recentVehicleLocation N2.latitudeDelta&gt;</b>		Integer for location Latitude value according 1.1 – OPTIONAL

Parameter	Defined Value/Definition
<recentVehicleLocation N2.longitudeDelta>	Integer for location Longitude value according 1.1 – OPTIONAL
<numberOfPassengers>	Integer - OPTIONAL
<optionalData.oid>	OID identifier in “dot notation” – OPTIONAL
<optionalData.data>	ASCII string – OPTIONAL

*Note:* The default values of the above parameters are the ones provided by the CEN example in related document 1.1.

### 4.3. Examples

Command	Responses
<b>AT+ASN1MSD=?</b>	OK
<b>AT+ASN1MSD=0</b>	ERROR
<b>AT+ASN1MSD=2</b>	+ASN1MSD: 015C0681D54970D65C3597CA0420C41464583ADE68A C52E9BB8413F149C07414FB414F6010180813E82181 823230 OK
<b>AT+ASN1MSD?</b>	+ASN1MSD: 1,1,1,0,1,1,"WM9","VDSVDS","Y","A123456",1, 0,0,0,1,0,123456789,173881200,41822520,14,1 0,-10,10,-20,2,1.2.125,"00FF" OK
<b>AT+ASN1MSD=0,0</b> <i>Note: Reset current values of mandatory parameters</i>	OK
<b>AT+ASN1MSD?</b>	+ASN1MSD: 0,0,0,0,0,0,"","","","",0,0,0,0,0,0,0,0,0, 10,-10,10,-20,2,1.2.125,"00FF" OK
<b>AT+ASN1MSD=0,1,1,1,"3",2,"WM9",, ,"Y",,"11"</b> <i>Note: Set of the ID, messageIdentifier, control, vehicleType, isowmi, isovisModelYear vehiclePropulsionStorageType parameters. Propulsion is both gasoline (0x1) and electric (0x10).</i>	OK



**SIERRA**  
WIRELESS®