Cellocator Wireless Communication Protocol - MiniTrack

Proprietary and Confidential

Version 1.0

Revised and Updated: September 25, 2019

Legal Notices

IMPORTANT

1. All legal terms and safety and operating instructions should be read thoroughly before the product accompanying this document is installed and operated.
2. This document should be retained for future reference.
3. Attachments, accessories or peripheral devices not supplied or recommended in writing by Pointer Telocation Ltd. May be hazardous and/or may cause damage to the product and should not, in any circumstances, be used or combined with the product.

General

The product accompanying this document is not designated for and should not be used in life support appliances, devices, machines or other systems of any sort where any malfunction of the product can reasonably be expected to result in injury or death. Customers of Pointer Telocation Ltd. using, integrating, and/or selling the product for use in such applications do so at their own risk and agree to fully indemnify Pointer Telocation Ltd. For any resulting loss or damages.

Warranty Exceptions and Disclaimers

Pointer Telocation Ltd. Shall bear no responsibility and shall have no obligation under the foregoing limited warranty for any damages resulting from normal wear and tear, the cost of obtaining substitute products, or any defect that is (i) discovered by purchaser during the warranty period but purchaser does not notify Pointer Telocation Ltd. Until after the end of the warranty period, (ii) caused by any accident, force majeure, misuse, abuse, handling or testing, improper installation or unauthorized repair or modification of the product, (iii) caused by use of any software not supplied by Pointer Telocation Ltd., or by use of the product other than in accordance with its documentation, or (iv) the result of electrostatic discharge, electrical surge, fire, flood or similar causes. Unless otherwise provided in a written agreement between the purchaser and Pointer Telocation Ltd., the purchaser shall be solely responsible for the proper configuration, testing and verification of the product prior to deployment in the field.

POINTER TELOCATION LTD.'S SOLE RESPONSIBILITY AND PURCHASER'S SOLE REMEDY UNDER THIS LIMITED WARRANTY SHALL BE TO REPAIR OR REPLACE THE PRODUCT HARDWARE, SOFTWARE OR SOFTWARE MEDIA (OR IF REPAIR OR REPLACEMENT IS NOT POSSIBLE, OBTAIN A REFUND OF THE PURCHASE PRICE) AS PROVIDED ABOVE. POINTER TELOCATION LTD. EXPRESSLY DISCLAIMS ALL OTHER WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, SATISFACTORY PERFORMANCE AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL POINTER TELOCATION LTD. BE LIABLE FOR ANY INDIRECT, SPECIAL, EXEMPLARY, INCIDENTAL OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOSS OR INTERRUPTION OF USE, DATA, REVENUES OR PROFITS) RESULTING FROM A BREACH OF THIS WARRANTY OR BASED ON ANY OTHER LEGAL THEORY, EVEN IF POINTER TELOCATION LTD. HAS BEEN ADVISED OF THE POSSIBILITY OR LIKELIHOOD OF SUCH DAMAGES.

Intellectual Property

Table of Contents

1 Introduction

1.1 About this Document

This document describes the unit wireless communication protocol structure implemented in Cellocator units. It describes every byte of the inbound/outbound messages, which can be sent/received by the unit over the air.

This document is comprised of the following main parts:

- Telemetry Channel (outbound messages initiated from the unit towards the server)
- Command Channel (inbound messages initiated from the server towards the unit)

Most communication flow scenarios between the unit and the server implement an acknowledge from the receiving side to the sending side. Some are implemented with a generic ACK (acknowledge) message, and some are implemented using other messages dedicated to the specific scenario.

A large portion of outbound messages from the unit to the server are initiated by the unit in a response to a certain trigger (e.g. GPIO activation, speed violation, etc.). Those messages are referred to as *events*. The unit supports the following kinds of events:

- **Logged Event:** If the condition for a specific logged event is met, the unit will create an event and store it into its non-volatile memory. The event will be sent to the server only during the GPRS session and will be deleted from the memory of the unit only after reception of an acknowledge from the server. Note that plain events will never be delivered by SMS.
- **Distress Event:** If the condition for a specific distress event is met, the unit will create a series of messages (session). The messages will be sent to the server immediately with the first available communication transport (during GPRS session – over IP, otherwise by SMS). The messages are not stored in the unit memory and if there is no cellular coverage when sending, the message will be lost. Distress events do not require an acknowledge from the server.

In addition to events, the unit supports a number of Power Modes:

Mode	GPS	Cell	Accelerometer	LED	BLE
Track	Yes	Yes	Yes	Yes	X
Listen	X	Yes	Yes	Yes	Yes
Sleep	X	POD (listen to SMS)	Yes	X	X
Hibernate	X	X	Yes	X	X

- During Hibernation mode (the most power efficient mode) the device does not communicate or perform tracking. It can be woken up from this mode via Accelerometer and Timers. In battery power devices it can also be awoken via power state or via GPIO events. When configured to operate in this mode, the device will hibernate and will not report any GPS related events, even if such events are enabled.

- During Sleep mode the device does not communicate and does not maintain a GPS position. It can be awakened by sensor or timer events, similarly to the Hibernation mode. Unlike when in Hibernate mode, the device will only go to sleep if no GPS related or other events that require the unit to stay awake are enabled. Note that some devices are still able to receive commands via SMS that will wake up the device.
- During Listen mode the device can communicate via IP/ SMS and all sensors are active, but the positioning function is off. As with Sleep mode, the device will only go to Listen (i.e. switch GPS off) if no GPS related events are enabled.
- During Track mode the device is fully operational, including BLE which is fully active. The unit can communicate via UART, SMS and IP and it will maintain its position.

1.2 Abbreviations

Abbreviation	Description
ACK	Acknowledge
DB	Database
FMS	Fleet Management System
OTA	Over the Air
PDU	Protocol Description Unit (common name for data SMS)
SMS	Short Message Service (GSM)
PTR	Pointer Telocation Ltd.
PSP	Pointer Serial Protocol, normally refers to a Car Alarm System interfacing through this protocol
NVM	Non-Volatile Memory
FW	Firmware
HW	Hardware

1.3 References

All the reference documents listed in the following table can be downloaded from the support section of the Pointer Website (www.pointer.com).

#	Reference	Description
1	Cellocator Programming Manual	This document describes the features supported by the Cellocator unit and provides details about the parameters of its configuration.
2	Cellocator Hardware Installation Guides	This document provides all necessary information for a technician who is involved in the installation of Cellocator units. It describes how to install and verify the proper functioning of the unit installation kit elements.

2 Telemetry Channel (Outbound Messages)

2.1 Overview

The telemetry channel comprises several types of messages, as described in the following sections:

- **Status/location Message (Message Type 0)** – a legacy message, which is sent by default, as a reply to a command or as the message of choice when reporting events or emergency situations.
- **Programming Data (Message Type 3)** – this message is sent as a reply to programming commands, or by request. It contains the new contents of the programmed block, which allows verification of the programming.
- **Modular Message (Message Type 9)** – this legacy modular message is designed to contain different types of data, such as commands, Cell ID, debug data, etc.

2.2 Status/Location Message (Message Type 0)

The message is used for reporting most of the basic unit events. It contains basic status data and location of the unit.

2.2.1 *Message Ingredients*

- Message Header
  - System Code – 4 bytes
  - Message Type – 1 byte
  - Unit ID – 4 bytes
  - Communication Control Field – 2 bytes
  - Message Numerator – 1 byte
- Unit Hardware Version – 1 byte
- Unit Firmware Version – 1 byte
- Protocol Version and Unit Functionalities – 1 byte
- Unit Status – 1 byte
- Current GSM Operator – 2 bytes
- Transmission Reason Specific Data – 1 byte
- Transmission Reason – 1 byte
- Unit Mode of Operation – 1 byte
- Unit I/O status – 4 bytes
- Analog Input Values – 4 bytes

- Mileage Counter (Odometer) – 3 bytes
- Multi-Purpose Field – 6 bytes
- Last GPS Fix – 2 bytes
- Service and Status – 1 byte
- Mode 1/2 – 2 bytes
- Number of Satellites Used – 1 byte
- Longitude – 4 bytes
- Latitude – 4 bytes
- Altitude – 4 bytes
- Ground Speed – 4 bytes
- Speed Direction (True Course) – 2 bytes
- Time and Date – 7 bytes
- Error Detection Code – 1 byte

2.2.2 *Byte-Aligned Table*

Byte	Description
1	System Code, byte 1 – ASCII "M"
2	System Code, byte 2 – ASCII "C"
3	System Code, byte 3 – ASCII "G"
4	System Code, byte 4 – ASCII "P"
5	Message Type (0)
6	Unit ID
7	
8	
9	
10	Communication Control Field
11	
12	Message Numerator (Anti-Tango™)
13	Unit Hardware Version
14	Unit Firmware Version

Byte	Description
15	Protocol Version and Unit Functionalities
16	Unit Status and Current GSM Operator (1st Nibble)
17	Current GSM Operator (2nd and 3rd Nibbles)
18	Transmission Reason Specific Data
19	Transmission Reason
20	Unit Mode of Operation
21	Unit I/O Status 1st byte
22	Unit I/O Status 2nd byte
23	Unit I/O Status 3rd byte
24	Unit I/O Status 4th byte
25	Current GSM Operator (4th and 5th Nibbles)
26	Analog Input 1 Value
27	Analog Input 2 Value
28	Analog Input 3 Value
29	Analog Input 4 Value
30	Mileage Counter (Odometer)
31	
32	
33	Multi-Purpose Field (Driver/Passenger/Group ID, PSP/Keyboard Specific Data, Accelerometer Status, SIM IMSI)
34	
35	
36	
37	
38	
39	Last GPS Fix
40	
41	Service and Status
42	Mode 1

Byte	Description
43	Mode 2
44	Number of Satellites Used
45	Longitude
46	
47	
48	
49	Latitude
50	
51	
52	
53	Altitude
54	
55	
56	
57	Ground Speed
58	
59	
60	
61	Speed Direction (True Course)
62	
63	UTC Time – Seconds
64	UTC Time – Minutes
65	UTC Time – Hours
66	UTC Date – Day
67	UTC Date – Month
68	UTC Date – Year (-2000) (e.g. value of 7 = year 2007)
69	
70	Error Detection Code (8-bit additive checksum, excluding system code)

Multiple byte fields are sent Intel style (i.e. least significant bytes sent first).

2.2.3 *Detailed Per-Field Specifications*

2.2.3.1 System Code

System code is a 4-byte value, which identifies the Cellocator system. The field is sent as the ASCII values of the letters "M", "C", "G", "P" (for IP messages) or "M", "C", "G", "S" (for SMS messages), in that order.

2.2.3.2 Message Type

Message type identifies the kind of the message. It allows the receiver to differentiate between different messages types, according to the value sent in this field.

Status/Location messages contain a value of 0 (zero) in the message type field.

2.2.3.3 Unit ID

This field contains a value that is uniquely assigned for every Cellocator unit during the manufacturing process. All messages sent by the same unit contain the same value in the Unit ID field.

2.2.3.4 Communication Control Field

This is a bitmapped field, providing information about the message and the situation in which it was originated.

First byte (10th):

		Multi-Purpose Field (Bytes 33-38) assignment		Message Source			Message Initiative
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Second byte (11th):

GSM Hibernation			Firmware Sub-Version				
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Message initiative

0 – Active transmissions (initiated by the unit, based on its logic and decisions)

1 – Passive responses (response to a command or a query message)

Message Source

0 – Direct message (not from memory)[1]

[1] The only exception is the "Transmission Reason 32 - IP changed/Connection up" message, which always requires ACK from server, even if it was sent as a direct message and not through memory.

1 – Message from memory (the unit tries to resend the message from the memory, until ACK from the server is received)

Multi-Purpose Field (Bytes 33-38) Assignment

These 2 bits, along with bit 7 in byte 41 of this message (Service and Status), define the data provided in bytes 33-38 of this message according to the following table:

Byte 41	Byte 10		Data in Bytes 33-38
Bit 7	Bit 5	Bit 4	
0	0	0	Driver ID/Keyboard Code (for AR units)
1	0	0	IMEI
X	X	X	IMSI (in Wake Up event (TR 202))

Note that the Communication Control Field is also sent in other (than 0) message types. In those message types, the Multi-Purpose Field (Bytes 33-38) Assignment indication is a "don't care".

Firmware Sub-Version

This field (5 bits) defines the firmware sub-version of the unit. The number of the Cellocator firmware is built from two parts: [Firmware version][Firmware sub-version], where firmware version usually defines the unit family and the sub-version defines the list of supported features.

For example, 30a:

- Version – 30
- Sub-Version – a (1)

Firmware Sub-Version Value (decimal)	Firmware Sub-Version Identifier
0	No identifier
1	a
2	b
3	c
…	…
26	z

For Minitrack - 200a

GSM Hibernation

0 – Unit is not in GSM hibernation

1 – Unit is in GSM hibernation (message sent during GSM peeking)

2.2.3.5 Message Numerator (Anti-Tango™)

The Message numerator field contains a value that is increased after every self-initiated generation of a message (in cases where an ACK from the server was received).

When the unit is reset/powered-up, this value is set to zero. This provides a way to chronologically sort incoming messages from a certain unit, in case an anachronistic communication medium is used.

NOTE: The unit assigns different message numerator sequences for the logged events and for real-time events. In passive transmission (reply to a command), the value in this field represents the number from the Command Numerator Field in an incoming command.

2.2.3.6 Unit Hardware Version

MiniTrack HW version: d48

2.2.3.7 Unit Firmware Version

This field defines the firmware version of the unit. The number of Cellocator firmware is built from two parts: [Firmware version][Firmware sub-version], where firmware version usually defines the unit family and the sub-version defines the list of supported features.

For example, 30a:

- Version – 30
- Sub-Version – a (1)

For Minitrack – 200a

2.2.3.8 Protocol Version

Sent as zero

2.2.3.9 Unit Status and Current GSM Operator (1st Nibble)

This is a bitmapped field, providing information about unit statuses and current GSM operator.

Current GSM Operator (PLMN), 1st nibble				zero	Correct Time	Home/ Roam Network	GPS Comm.
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

GPS Comm.

0 – Communication with GPS is available

1 – Communication with GPS is not available

Home/Roam Network

0 – Home network

1 – Roam network

Correct Time

0 – Valid time stamp

1 – Invalid/estimated time stamp

Current GSM Operator

The current GSM Operator (PLMN) is represented as a 5-character hexadecimal number. After conversion into decimal it represents the MCC-MNC of a cellular operator (country code + network number). The 5 PLMN nibbles (nibble for each character) are provided in the following places:

Nibble 1	Nibble 2	Nibble 3	Nibble 4	Nibble 5
Byte 16 (4MSbits)	Byte 17		Byte 25	

2.2.3.10 Current GSM Operator (2nd and 3rd Nibbles)

Current GSM Operator (PLMN), 2nd Nibble				Current GSM Operator (PLMN), 3rd Nibble			
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

2.2.3.11 Transmission Reason Specific Data

Additional information related to the transmission reason (specified in byte 19):

<table>
<tr><th>Transmission Reason</th><th colspan="2">Transmission Reason Specific Data Description</th></tr>
<tr><td rowspan="3">8 Towing
Not Supported</td><td>0</td><td>Location change detected during ignition off</td></tr>
<tr><td>1</td><td>Towed mode start</td></tr>
<tr><td>2</td><td>Towed mode stop</td></tr>
<tr><td>15
Crash detection</td><td colspan="2">
<table>
<tr><td colspan="2">Reserved</td><td>Light crash event</td><td>Heavy crash event</td><td colspan="4">Peak RMS value of the impact in 1g resolution minus 1g
(16g=0xF, 1g=0x0)</td></tr>
<tr><td>Bit 7</td><td>Bit 6</td><td>Bit 5</td><td>Bit 4</td><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Transmission Reason</th><th>Transmission Reason Specific Data Description</th></tr>
<tr><td>191
Geo hotspot violation</td><td>
<table>
<tr><td>Direction
0 – exit from hot spot
1 – entry to hot spot</td><td colspan="7">The index of the geo-fence</td></tr>
<tr><td>Bit 7</td><td>Bit 6</td><td>Bit 5</td><td>Bit 4</td><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr>
</table>
</td></tr>
<tr><td>206
Jamming detection
Future Feature</td><td>
<table>
<tr><td colspan="7"></td><td>GSM jamming state
0 – GSM jamming detection start
1 – GSM jamming detection end</td></tr>
<tr><td>Bit 7</td><td>Bit 6</td><td>Bit 5</td><td>Bit 4</td><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr>
</table>
</td></tr>
<tr><td>207
Radio off mode
Future Feature</td><td>
<table>
<tr><td colspan="4">Spare</td><td>Airplane Mode</td><td>Early
Radio Off Event</td><td>GPS Status
0 – Off
1 – On</td><td>Modem Status
0 – Off
1 – On</td></tr>
<tr><td>Bit 7</td><td>Bit 6</td><td>Bit 5</td><td>Bit 4</td><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr>
</table>
</td></tr>
</table>

2.2.3.12 Transmission Reason

This field contains the reason for the message transmission. Note that this value is valid only for self-initiated active transmissions, i.e. transmissions that the unit generated because of its logics, in contrast to reply transmissions. Reply transmissions contain the last transmission reason that was used.

Value	Transmission Reason Description	Description

<table>
<tr><th>Value</th><th>Transmission Reason Description</th><th>Description</th></tr>
<tr><td>8</td><td>Location change detected during ignition off (Towing)
Future Feature</td><td>Towing detection logic is activated in Engine Off mode irrespective of Hibernation mode. The unit will store the coordinates upon Stop alert generation (only if the location considered as valid).
During parking (continually, upon GPS peek, as per hibernation mode settings; the GPS peek can also be activated upon movement detection by accelerometer) the unit will examine its location and speed. If the location changes from the Journey Stop or speed are detected while ignition switch is off – the towing alert will be triggered (See also Transmission Reason 25).</td></tr>
<tr><td>11</td><td>Communication idle</td><td>This event is being triggered if no other event/message was generated (actively passively) during the defined time period.
This algorithm is frequently used as a "Heart Bit" of the unit, a kind of "Keep Alive" messaging.</td></tr>
<tr><td>15</td><td>Crash detection</td><td>The unit’s accelerometer is used as a source for crash detection and reporting.
The feature includes 2 separate RMS based thresholds, one for light crash and one for heavy crash. The detection works both when the Ignition is on and off.
Cellocator FW samples the Accelerometer Data at 100 Hz rate.
Only 3 consequential samples which violate the crash threshold are considered a crash.
Transmission Reason Specific Data (previous byte)
<table>
<tr><td colspan="2">Reserved</td><td>Light crash event</td><td>Heavy crash event</td><td colspan="4">Peak RMS value of the impact in 1g resolution minus 1g
(16g=0xF, 1g=0x0)</td></tr>
<tr><td>Bit 7</td><td>Bit 6</td><td>Bit 5</td><td>Bit 4</td><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr>
</table></td></tr>
<tr><td>25</td><td>Speed detected during ignition off (Towing)</td><td>Towing detection logic is activated in Engine Off mode irrespective of Hibernation mode. The unit will store the coordinates upon Stop alert generation (only if the location is considered as valid).
During parking (continually, upon GPS peek, as per hibernation mode settings; the GPS peek can also be activated upon movement detection by accelerometer) the unit will examine its location and speed. If the location changes from the Journey Stop or speed are detected while ignition switch is off, the towing alert will be triggered (see also Transmission Reason 8- TR 8 Not Supported).</td></tr>
</table>

Value	Transmission Reason Description	Description
31	Reply to command	Value 31 will be sent upon reply to any OTA command type 0 (Status request, reset, output activation etc).
32[2]	IP changed/connection up	Will be reported upon renewal of IP connection. Requires ACK (in order to enable log upload), but not logged.
33	GPS navigation start	Will be reported upon acquisition of valid GPS. See also event 41
34	Over speed start	If velocity of the vehicle is higher than pre-programmed threshold, for longer than programmed in a corresponding parameter, the unit will consider over-speeding start. See also over-speeding End event (42) below.
35	Idle speed start	Idle Speeding refers to a situation when the vehicle is parking with a working engine. If velocity of the vehicle is lower than the Idle Speed threshold, for longer than programmed in a pre-programmed timeout (from ignition on or from higher speed), and the Ignition is On, the unit will consider Idle Speeding start. See also Idle-speeding End event (43) below.
36	Distance event	This event is generated upon passing a preprogrammed distance from a last distance event or from Ignition On.
37	Accelerometer movement detection start	This event is generated when the accelerometer detects engine on pattern (see also event 45).
38	GPS factory reset (maintenance update)	This maintenance event is generated when the unit detects a GPS malfunction and performs its factory reset.
41	GPS navigation end	Will be reported upon loss of GPS signal. See also event 33.
42	Over speed end	During over-speeding, if velocity of the vehicle falls lower than corresponding threshold, the unit will consider over-speeding end. See over-speeding start event (34) above.
43	Idle speed end	During idle-speeding, if velocity of the vehicle exceeds the corresponding threshold, the unit will consider idle-speeding end. See idle-speeding start event (35) above.

[2] Always requires acknowledge from server, even if it was sent as a direct message and not through memory.

Value	Transmission Reason Description	Description
44	Timed event	This is a periodical update during Ignition On model; it should be possible to define between 1 msg/sec to 1msg/hour.
45	Accelerometer movement detection stop	This event is generated when the accelerometer stops detecting engine on pattern (see also event 37).
46	Driver authentication update (infrastructure)	(infrastructure only) Not supported by the first version of MiniTrack.
47	Driving without authentication (infrastructure)	(infrastructure only) Not supported by the first version of MiniTrack.
48	Door (Input 1) inactive	This event is generated upon detection of Input 1 state change to Inactive. See also event 64.
49	Shock (Input 2) inactive	This event is generated upon detection of Input 2 state change to Inactive. See also event 65.
53	Driving (logical) stop	This event is generated upon stop detection (logical state, normally when both Ignition and Accelerometer detects lack of movement for pre-programmed timeout). See also event 69.
63	Ignition input – inactive	This event is generated upon detection of Ignition physical input change to Active. See also event 79.
64	Door (Input 1) active	This event is generated upon detection of Input 1 state change to Active. See also event 48.
65	Shock (Input 2) active	This event is generated upon detection of Input 2 state change to Active. See also event 49.
69	Driving (logical) start	This event is generated upon start detection (logical state, normally when both Ignition and Accelerometer detects movement for pre-programmed time period). See also event 53.
79	Ignition input active	This event is generated upon detection of Ignition physical input change to Active. (See also event 63).
80	Main power disconnected	This event generated upon main power disconnected (see also event 87).
81	Main power low level	This event generated upon voltage level is low. This might trigger moving to a lower power mode or less frequent reporting. (See also event 88).

<table>
<tr><th>Value</th><th>Transmission Reason Description</th><th>Description</th></tr>
<tr><td>82</td><td>Backup battery disconnected</td><td>This event generated upon internal battery disconnected</td></tr>
<tr><td>83</td><td>Backup battery low level</td><td>This event generated upon internal battery voltage level is low. This might trigger moving to a lower power mode or less frequent reporting.
(See also event 90).</td></tr>
<tr><td>87</td><td>Main power connected</td><td>This event is unconditionally generated upon a power up (see also event 80).</td></tr>
<tr><td>88</td><td>Main power high level</td><td>This event generated upon voltage level higher than considered normal (See also event 81).</td></tr>
<tr><td>89</td><td>Backup battery connected</td><td>This event generated upon internal battery connected</td></tr>
<tr><td>90</td><td>Backup battery high level</td><td>This event generated upon internal battery with voltage higher than considered normal.</td></tr>
<tr><td>191</td><td>Geo Fence: hotspot violation</td><td>This event is triggered on entry and on exit to/from one of 30 geo-fenced zones.

Transmission Reason Specific Data (previous byte)
<table>
<tr><td>Direction
0 – exit from hot spot
1 – entry to hot spot</td><td colspan="7">The index of the geo-fence</td></tr>
<tr><td>Bit 7</td><td>Bit 6</td><td>Bit 5</td><td>Bit 4</td><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr>
</table></td></tr>
<tr><td>202</td><td>Wake Up event (for auto-assignment SIM number to Unit ID)</td><td>In most of the cases the unit is sent to the installation site without a SIM card. The SIM card is being inserted at the installation site.
In order to create an automatic association of the unit with the SIM number in the CCC application, the unit initiates a real-time message containing IMSI in the Multi-Purpose Field using SMS or GPRS (as per configuration).</td></tr>
<tr><td>203</td><td>Pre-hibernation event</td><td>This event is generated upon entrance to hibernation mode.</td></tr>
</table>

<table>
<tr><th>Value</th><th>Transmission Reason Description</th><th>Description</th></tr>
<tr><td>206</td><td>Jamming detection
Future Feature</td><td>This event is logged upon detection of GSM jamming start end end.
Transmission Reason Specific Data (previous byte)
<table>
<tr><td colspan="7"></td><td>GSM jamming state
0 – GSM jamming detection start
1 – GSM jamming detection end</td></tr>
<tr><td>Bit 7</td><td>Bit 6</td><td>Bit 5</td><td>Bit 4</td><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr>
</table></td></tr>
<tr><td>247</td><td>Finish mode (self re-flash process update)</td><td>Firmware upgrade successfully accomplished, the unit resets itself.</td></tr>
</table>

2.2.3.13 Unit Mode of Operation

The functioning of the unit can be generalized as a finite state machine model, with a few "stages" of operation. The "current stage" is referred to as "unit mode", or "mode of operation", as per the following:

Unit Mode Value	Unit Mode Description
0x00	Standby Engine On
0x01	Standby Engine Off

2.2.3.14 Unit I/O Status

The unit is provided with many I/Os (inputs/outputs). Each I/O can be "high" or "low" at a given moment. The I/O status field is a bitmapped representation of the I/Os physical levels. Note that the I/Os that have been configured to be inverted will affect the application but will not be shown in this field, as it only represents the raw physical signals read from the HW.

1st Byte of I/O Status

MiniTrack			Driving Status (physical ignition or accelerometer based)				Shock (Input 2)	Door (Input 1)
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Note: Driving Status (bit 5) provides indication if the unit is in logical Ignition On/Off, according to the configuration of the detected source (physical ignition or accelerometer). It will indicate "1" when logical Ignition On is detected, and "0" when logical Ignition Off is detected.

2nd Byte of I/O Status

MiniTrack	Ignition port status	Accelerometer status						
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Notes: Accelerometer Status (bit 6) provides indication if the accelerometer has detected Ignition On/Off, **regardless** of the configuration of the detected source (physical ignition or accelerometer). It will indicate "1" when accelerometer Ignition On is detected, and "0" when accelerometer Ignition Off is detected. Ignition Port Status (bit 7) provides indication if the physical ignition input is high/low, **regardless** of the configuration of the detected source (physical ignition or accelerometer). It will indicate "1" when the ignition input is high, and "0" when the ignition input is low.

3rd Byte of I/O Status

CR300/ CR300B					GPS Power	Grad. Stop		-
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

4th Byte of I/O Status

CR300/ CR300B	Charger status		Standard Immobilizer (Output 2)		Blinkers (Output 1)			
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

2.2.3.15 Current GSM Operator (4th and 5th Nibbles)

Current GSM Operator (PLMN), 4th Nibble				Current GSM Operator (PLMN), 5th Nibble			
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

2.2.3.16 Analog Inputs

The unit may handle up to 4 analog inputs. These inputs are multiplexed and sent as 8-bit samples each. The allocation of measurements to the bytes of the message is configurable (PL addresses 1620-1623).

For the Cello/CR300 unit:

Field name	Default value	Byte number in the message
1st analog measurement	9 (Vin)	26
2nd analog measurement	6 (Vbat)	27
3rd analog measurement	7 (NTC)	28
4th analog measurement	RSSI	

2.2.3.17 Mileage Counter (Odometer)

The unit is provided with a distance accumulator feature. The unit counts distance "base units" programmed in the PL (which might be hardcoded to 100m).

By synchronizing the accumulator value with the vehicle odometer reading and setting the distance base units to one kilometer/mile, this counter provides the ability to remotely read the vehicle odometer. The programming and synchronizing is only needed once – during the installation.

The mileage counter field contains the current 24-bit value of this accumulator.

2.2.3.18 Multi-Purpose Field (Bytes 33-38)

This field may carry different information as per bits 4, 5 in Communication Control Field (byte 10) and bit 7 in Service and Status (byte 41):

Byte 41	Byte 10		Data in Bytes 33-38
Bit 7	Bit 5	Bit 4	
0	0	0	Driver ID
1	0	0	IMEI
X	X	X	IMSI (in Wake Up event (TR 202))

Driver ID/Passenger ID/Group ID Code Update

The unit can provide 6 bytes of the last received Dallas button in every message if that feature is enabled in the PL (Mask of Authentication Events).

If no Dallas code is received since the initiation of the last Start Event, this field will be 0.

IMEI

IMEI is sent on bytes 33-38 with its 2 MS-Bits sent in bits 5, 6 in byte 41 of this message (Service and Status).

IMEI is defined as 15 decimal digits. Converting the maximal IMEI number 999999999999999 to hexadecimal we get: 38D7EA4C67FFF. The maximal number will occupy 50 bits which will be sent as follows:

0x03	0xFF	0x7F	0xC6	0xA4	0x7E	0x8D
Byte 41, bits 5, 6	Byte 33	Byte 34	Byte 35	Byte 36	Byte 37	Byte 38

Note: For CDMA devices, the IMEI is replaced with MEID, which is 18 decimal digits long. Thus, MEID will not be transmitted in these bytes (only in Type 9, sub data 0x12).

IMSI

In case of a Wake Up event (TR 202), the unit reports the 12 first characters of the SIM IMSI converted to hex (Little Endian).

The IMSI number consists of up to 15 numerical characters (0-9). An IMSI consists of a 3-digit mobile country code (MCC, which is not reported by the Cellocator Protocol) and a variable length national mobile station identity (NMSI).

The NMSI consists of two variable length parts: the mobile network code (MNC) and the mobile station identification number (MSIN). A Class 0 IMSI is 15 digits in length. A Class 1 IMSI is less than 15 digits in length.

Example: 425020315229000 (Cellcom IL)

MCC	425	Israel
MNC	02	Cellcom IL
MSIN	0315229000	

The Hex value received in bytes 33-38:

Value (hex)	00	5A	16	0F	03	02
Location	Byte 33	Byte 34	Byte 35	Byte 36	Byte 37	Byte 38

Conversion table:

In wireless protocol (big-endian)	00	5A	16	0F	03	02
HEX values (little-endian)	02	03	0F	16	5A	00
DEC values	02	03	15	22	90	00
NMSI (MNS + MSIN)	020315229000					

2.2.3.19 Last GPS Fix

This field provides a timestamp to indicate when the GPS was last in navigation mode.

Day of Month					Hours					Minutes					
Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 40								Byte 39							

NOTE: The easiest way to define if the GPS data in the message is valid and updated, or historical, is to compare between the time of the timestamps and UTC time (see below).

2.2.3.20 Service and Status

MSB of Multi-Purpose field (bytes 33-38) assignment (with bits 4, 5 of byte 10)	IMEI Bit 49	IMEI Bit 48					
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

2.2.3.21 MODE 1 and Mode 2

These fields are generated by the GPS and transparently monitored in the outgoing message from the unit. The fields define the validity of GPS data in the message.

The unit considers the valid fix according to the "Enable Tight GPS PMODE Filter" parameter (address 509, bit 7):

- If "Enable Tight GPS PMODE Filter" is enabled, the unit considers the GPS data as valid only if Mode 1 = 3 or 4 AND Mode 2 = 2
- If "Enable Tight GPS PMODE Filter" is disabled, the unit considers the GPS data as valid only if Mode 1 =2, 3, 4, 5 and 6

2.2.3.22 Number of Satellites Used

Number of satellite measurements used for current position fix. Possible values are 0 to 20 (GNSS modules).

2.2.3.23 Longitude, Latitude

Longitude and latitude coordinates of current position fix. Both coordinates are sent as 32-bit signed integers, representing the coordinates in 10^{-8} radian resolution. Possible values are -Π to +Π for longitude, or – Π/2 to + Π/2 for latitude. The coordinates refer to WGS-84 map datum and ellipsoid.

2.2.3.24 Altitude

Altitude of current position fix. Represented as a 32-bit signed integer, in 10^{-2} meter resolution (altitude is represented in centimeters).

2.2.3.25 Ground Speed

Current speed (absolute value of the vector). Represented as a 32-bit unsigned integer, in 10^{-2} meter/sec resolution (speed is represented in centimeters/sec).

2.2.3.26 Heading/Speed Direction (True Course)

Direction (angle) of the speed vector. Represented as 16-bit unsigned integer, in 10^{-3} radian resolution. Possible values are 0 to 2Π.

2.2.3.27 System Time

Universal coordinated time of the position fix, represented in seconds (0-59), minutes (0-59) and hours (0-23).

Note that the system time and date fields are monitoring system time, based on the internal timer of the unit. The internal timer synchronizes with GPS time when the GPS fix is considered as valid (or always as per configuration flag).

2.2.3.28 System Date

Universal coordinated date of the position fix, represented in days (1-31), months (1-12) and years (1980-2079).

Note that the system time and date fields are monitoring system time, based on the internal timer of the unit. The internal timer synchronizes with GPS time when the GPS fix is considered as valid (or always as per configuration flag).

2.2.3.29 Error Detection Code

The error detection code (checksum) is a last byte of sum of all bytes in a message, excluding the 4 bytes of System Code and the Error Detection Code itself.

Example:
The message:
*4D434750*0006000000081A020212040000002100623000000006B00E100000000000000000000E5A
100040206614EA303181A57034E1200000000000000001525071403D607*CS*

Calculation of the CS=>

00+06+00+00+00+08+1A+02+02+12+04+00+00+00+21+00+62+30+00+00+00+6B+00+E1+
00+00+00+00+00+00+00+00+00+00+E5+A1+00+04+02+06+61+4E+A3+03+18+1A+57+03+
4E+12+00+00+00+00+00+00+00+00+15+25+07+14+03+D6+07=0x749
=>CS=0x49

2.3 Programming Data (Message Type 3)

This message is sent as a reply to programming commands, or by request. It contains the new contents of the programmed block.

2.3.1 *Message Ingredients*

- Message header
  - System Code – 4 bytes
  - Message Type – 1 byte
  - Unit ID – 4 bytes
  - Communication Control Field – 2 bytes
  - Message Numerator – 1 byte

- Spare – 1 byte
- Block Code – 1 byte
- Block Data – 16 bytes
- Error Detection Code – 1 byte

2.3.2 *Byte-Aligned Table*

Byte	Description
1	System Code, byte 1 – ASCII "M"
2	System Code, byte 2 – ASCII "C"
3	System Code, byte 3 – ASCII "G"
4	System Code, byte 4 – ASCII "P"
5	Message Type (3)
6	Unit ID
7	
8	
9	
10	Communication Control Field
11	
12	Message Numerator (Anti-Tango™)
13	Spare
14	Block Code
15-30	Block Data
31	Error Detection Code (8-bit additive checksum, excluding system code)

2.3.3 *Detailed Per-Field Specifications*

2.3.3.1 System Code

Refer to Section 2.2.3.1

2.3.3.2 Message Type

Programming Data messages contain a value of 3 (three) in the message type field.

2.3.3.3 Unit ID

Refer to Section 2.2.3.3

2.3.3.4 Communication Control Field

Refer to Section 2.2.3.4

2.3.3.5 Message Numerator (Anti-Tango™)

Refer to Section 2.2.3.5

2.3.3.6 Block Code

OTA (over the air) parameter programming is done in blocks. The entire parameter memory is partitioned to 16-byte long blocks. Each of those blocks is identified with a block code. The block code field contains the code of the block whose data is sent in this message (in the block data field).

2.3.3.7 Block Data

Contains the actual data programmed in the specified block of the parameter memory.

2.4 Modular Message (Message Type 9)

The modular data packet is designed to provide different data types in the same message.

2.4.1 *Message Ingredients*

- Message Header
  - System Code – 4 bytes
  - Message Type – 1 byte
  - Unit ID – 4 bytes
  - Communication Control Field – 2 bytes
  - Message Numerator – 1 byte
- Packet Control Field – 1 byte
- Message Length – 1 byte
- First Sub-Data Type – 1 byte
- First Sub-Data Length – 1 byte
- First Sub-Data variable length, depends on Data Type
- ……
- Nth Sub-Data Type – 1 byte
- Nth Sub-Data Length – 1 byte
- Nth Sub-Data– variable length, depends on Data Type N
- Error Detection Code – 1 byte

2.4.2 *Byte-Aligned Table*

Byte	Description
1	System Code, byte 1 – ASCII "M"
2	System Code, byte 2 – ASCII "C"
3	System Code, byte 3 – ASCII "G"
4	System Code, byte 4 – ASCII "P"
5	Message Type (9)
6	Unit ID
7	
8	

Byte	Description
9	
10	Communication Control field
11	
12	Message Numerator
13	Packet Control Field
14	Length (of the modules section - not including the checksum)
15	First Sub-data Type
16	First Sub-data Length
17	First Sub-data The Data
…	…
	Nth Sub-data Type
	Nth Sub-data Length
	Nth Sub-data The Data
Last Byte	Error Detection Code (8-bit additive checksum, excluding system code)

2.4.3 *Detailed Per-Field Specifications*

2.4.3.1 Packet Control Field

Direction	Out of space indication	Unused					
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Direction

0 – Data from the unit

1 – Request (unit-bound)

Out of Space Indication

0 – All the requested data present in the message

1 – Some Sub-Data was not returned due to data size

2.4.3.2 Length

This field includes the number of data bytes with their types and lengths. It includes the number of bytes from byte 15 to the byte of the checksum, which is not included.

2.4.4 *Outbound Sub-Data Types Table*

Code (Hex)	Function
0x01	Firmware Platform Manifest (for FOTA reflash)
0x09	Neighbor List of the Serving GSM Cell
0x0A	Maintenance Server Platform Manifest
0x0C	3G Cell ID Data

2.4.5 *Firmware Platform Manifest*

TBD for MINITRACK

This sub-data is generated as a reply to Firmware Platform Manifest Request (0x01).

Byte	Description
0	Sub-data type (0x01)
1	Length – 18
2	Processor family identifier 0x01 – PIC18Fx520/620/720 0x02 – PIC18Fx621/525 0x03 – PIC18Fx527/622/627/722 (x=6/8) 0x04 – ARM Cortex M3 F10x 0x05 – ARM Cortex M3 L15x 0x07 – STM32F101RDT6 0x08 – STM32F103RFT6 0x09 – STM32F429IGH6 0x0A – STM32F103VET7 0x0B – STM32L151VDT6 0x20 - MediaTek 2503A (ARM7EJ-S)
3	Hardware interface and peripherals identifier 0x01 – 40/44 pin micro, peripherals as per family 0x02 – 64 pin micro, peripherals as per family 0x03 – 80 pin micro, peripherals as per family 0x04 – 64 pin STM32F101RCT6 0x05 – 64 pin STM32F103RDT6 0x06 – 64 pin STM32L151RDT6 0x07 – 176 pin micro, peripherals as per family 0x20 - TFBGA packaging, peripherals per family

Byte	Description
4-5	Size of program memory (in 1024 bytes units) (LSB)
	Size of program memory (in 1024 bytes units) (MSB)
6-7	Size of volatile memory (Divided by 128 bytes and rounded up/down to closest integer) (LSB)
	Size of volatile memory (Divided by 128 bytes and rounded up/down to closest integer) (MSB)
8-9	Size of internal non-volatile memory (Divided by 128 bytes and rounded up/down to closest integer) (LSB)
	Size of internal non-volatile memory (Divided by 128 bytes and rounded up/down to closest integer) (MSB)
10-11	Size of external non-volatile memory (in 1024 bytes units) (LSB)
	Size of external non-volatile memory (in 1024 bytes units) (MSB)
12	External non-volatile memory type 0x01 – I2C generic NVM (most EEPROMs). 0x02 – SPI generic NVM. 0x03 – Adesto Rev. E 0x04 – SPI N25Q NVM 0x05 – SPI MX25L6433F 0x20 - SPI MX25U6435E(Micronix)
13	Hardware Version 0x30 – Minitrack 48 See Unit Hardware Version
14-15	Reprogramming facility identifier (LSB) Depends on HW/FW variant
	Reprogramming facility identifier (MSB) Depends on HW/FW variant
16-17	Script language version (LSB) (0x01)
	Script language version (MSB) (0x00)

Byte	Description
18-19	Current Firmware ID (LSB) Note that this is in fact not a descriptor of the firmware platform per se, but rather a descriptor of the actual firmware running on the platform. However, it is a valuable field when a re-flash is considered.
	Current Firmware ID (MSB) Note that this is in fact not a descriptor of the firmware platform per se, but rather a descriptor of the actual firmware running on the platform. However, it is a valuable field when a re-flash is considered.

2.4.6 *Authentication Table Update*

Not supported, currently sends as zeros.

2.4.7 *Neighbor list of the Serving GSM Cell*

This sub-data is sent:

- Passively, as a reply to Cell ID Request (0x09).
- Actively, if enabled in unit's configuration, separately for home and roam GSM networks (addresses 201 and 203 respectively, bits 0, 1, 3 and 4).

Byte	Description
0	Sub-Data Type (0x09)
1	Length – 53
2	Spare (0x00)
3	seconds (0-59)
4	minutes (0-59)
5	hours (0-23)
6	day (1-31)
7	month (1-12)
8	Year (Current Year minus 2000 (e.g. value of 7 = year 2007))
9	Serving Cell BSIC (Base Station Identification Code)

Byte	Description
10	Serving Cell LAC (LSB) (Localization Area Code)
11	Serving Cell LAC (MSB) (Localization Area Code)
12	Serving Cell ID (LSB)
13	Serving Cell ID (MSB)
14	Serving Cell Power (Received signal strength in dBm (hex). The sign is not saved, this value is always representing a negative number)
15	Neighbor Cell 1 BSIC
16	Neighbor Cell 1 LAC (LSB)
17	Neighbor Cell 1 LAC (MSB)
18	Neighbor Cell 1 Cell ID (LSB)
19	Neighbor Cell 1 Cell ID (MSB)
20	Neighbor Cell 1 Power
…	…
45	Neighbor Cell 6 BSIC
46	Neighbor Cell 6 LAC (LSB)
47	Neighbor Cell 6 LAC (MSB)
48	Neighbor Cell 6 Cell ID (LSB)
49	Neighbor Cell 6 Cell ID (MSB)
50	Neighbor Cell 6 Power
	Zero Padding to complete the 55 bytes assigned for single event (if it's a logged event, i.e. sent actively)

2.4.8 *Maintenance Server Platform Manifest*

TBD FOR MINITRACK

Periodically (or upon server command) the unit connects to a maintenance server in order to check for the latest firmware and/or programming update. Auto connection to the maintenance server can be enabled upon power up and upon firmware upgrade.

Upon connection the unit generates a sub-data which is described below.

If the unit cannot establish a connection to the maintenance server while the GPRS is available, it uses the dial up retry algorithm defined in the NVM Allocation (Anti-Flooding). If all the retries fail, the unit ceases to try and reconnects to an operational server (instead of entering Anti-Flooding, as it would do while connected to an operational server).

Byte	Description
0	Sub-data type (0x0A)
1	Length – 34
2	Processor family identifier 0x01 – PIC18Fx520/620/720 0x02 – PIC18Fx621/525 0x03 – PIC18Fx527/622/627/722 (x=6/8) 0x04 – ARM Cortex M3 F10x 0x05 – ARM Cortex M3 L15x 0x07 – STM32F101RDT6 0x08 – STM32F103RFT6 0x09 – STM32F429IGH6 0xA – STM32F103VET7 0x0B – STM32L151VDT6 0x20 - MediaTek 2503A (ARM7EJ-S)
3	Hardware interface and peripherals identifier 0x01 – 40/44 pin micro, peripherals as per family 0x02 – 64 pin micro, peripherals as per family 0x03 – 80 pin micro, peripherals as per family 0x04 – 64 pin STM32F101RCT6 0x05 – 64 pin STM32F103RDT6 0x06 – 64 pin STM32L151RDT6 0x07 – 176 pin micro, peripherals as per family 0x20 - TFBGA packaging, peripherals per family

Byte	Description
4-5	Size of program memory (in 1024 bytes units) (LSB)
	Size of program memory (in 1024 bytes units) (MSB)
6-7	Size of volatile memory (Divided by 128 bytes and rounded up/down to closest integer) (LSB)
	Size of volatile memory (Divided by 128 bytes and rounded up/down to closest integer) (MSB)
8-9	Size of internal non-volatile memory (Divided by 128 bytes and rounded up/down to closest integer) (LSB)
	Size of internal non-volatile memory (Divided by 128 bytes and rounded up/down to closest integer) (MSB)
10-11	Size of external non-volatile memory (in 1024 bytes units) (LSB)
	Size of external non-volatile memory (in 1024 bytes units) (MSB)
12	External non-volatile memory type 0x01 – I2C generic NVM (most EEPROMs). 0x02 – SPI generic NVM. 0x03 – Adesto Rev. E 0x04 – SPI N25Q NVM 0x05 – SPI MX25L6433F 0x20 - SPI MX25U6435E (Micronix)
13	Hardware Version 0x30 – Minitrack 48 See: Unit Hardware Version
14-15	Reprogramming facility identifier (LSB) Depends on HW/FW variant
	Reprogramming facility identifier (MSB) Depends on HW/FW variant
16-17	Script language version (LSB) (0x01)
	Script language version (MSB) (0x00)

<table>
<tr><th>Byte</th><th>Description</th></tr>
<tr><td rowspan="2">18-19</td><td>Current Firmware ID (LSB)
Note that this is in fact not a descriptor of the firmware platform per se, but rather a descriptor of the actual firmware running on the platform. However, it is a valuable field when a re-flash is considered.</td></tr>
<tr><td>Current Firmware ID (MSB)
Note that this is in fact not a descriptor of the firmware platform per se, but rather a descriptor of the actual firmware running on the platform. However, it is a valuable field when a re-flash is considered.</td></tr>
<tr><td rowspan="2">20-21</td><td>Current PL ID (LSB)
Infrastructure only, currently not supported</td></tr>
<tr><td>Current PL ID (MSB)
Infrastructure only, currently not supported</td></tr>
<tr><td>22-29</td><td>International mobile subscriber identity of the SIM (IMSI)
Reference to GSM 07.07, 15 chars maximum</td></tr>
<tr><td>30-32</td><td>Modem's firmware revision
From FW version 33x and later – 0x00
For FW versions older than 33x:
<table>
<tr><th>Byte</th><th>Description</th></tr>
<tr><td>30</td><td>Reserved (0)</td></tr>
<tr><td>31</td><td>Modem Revision ID, as presented in the table below</td></tr>
<tr><td>32</td><td>Modem Type Extension
(Extra byte, additional to the 3 MSBits in the hardware byte of message type 0)</td></tr>
</table></td></tr>
<tr><td>33</td><td>Maintenance Configuration
<table>
<tr><td colspan="6">Spare</td><td>Firmware Upgrade Enabled

0 - Disabled
1 - Enabled</td><td>Programming Enabled

0 - Disabled
1 - Enabled</td></tr>
<tr><td>Bit 7</td><td>Bit 6</td><td>Bit 5</td><td>Bit 4</td><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr>
</table></td></tr>
</table>

Byte	Description
34 35	Release Candidate Revision ID Little Endian 16 bit representing the Release Candidate SVN revision: <table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Formal Release</td></tr><tr><td>1-65535</td><td>The version is a Release Candidate. The 2 bytes represents the SVN revision number: Example: If the hex file name is F000_...._RC540.hex the resulted The binary value representing the decimal RC540 is 0x21C in little Endian.</td></tr></table>

Modem Revision ID

ID (Dec)	Revision	Modem
0	Unknown	All (Used also in Nano from FW version 34d and on).
83-255	Reserved	

2.4.9 *3G Cell ID Data*[3]

This sub-data is sent:

- Passively, as a reply to Cell ID data request (0x09). In this case the sub-data will be sent using the same communication transport as the request.
- Actively, if enabled in unit configuration, separately for home and roam GSM networks, on address 201 and 203 respectively, bits 0, 1, 3 and 4.

Byte	Description
0	Sub-Data Type (0x0C)
1	Length – 53
2	Spare

[3] Supported for 3G variants only

Byte	Description
3	seconds (0-59)
4	minutes (0-59)
5	hours (0-23)
6	day (1-31)
7	month (1-12)
8	Year (Current Year minus 2000 (e.g. value of 7 = year 2007))
9	MCC (LSB) (Mobile Country Code, Decimal, 200-901)
10	MCC (MSB) (Mobile Country Code, Decimal, 200-901)
11	MNC (LSB) (Mobile Network Code, Decimal, 0-999)
12	MNC (MSB) (Mobile Network Code, Decimal, 0-999)
13	LAC (LSB) (Localization Area Code)
14	LAC (MSB) (Localization Area Code)
15	RSCP (Received Signal Code Power - Energy after processing with gain from coding, which is equivalent to RSSI [RSCP=RSSI+EC/No]; dBm units, (-120)≤RSCP≤(-25))
16	Cell ID (Expanded 3G Cell ID (RNC + Cell ID + possible sector))
17	
18	
19	
20	Power (Received signal strength (hex); the sign is not saved, this value is always representing a negative number; dBm units)

Byte	Description
21	AcT (Access Technology) 0 – GSM 2 – UTRAN
22	PSC (LSB) (Primary Scrambling Code, Decimal, 0-65535)
23	PSC (MSB) (Primary Scrambling Code, Decimal, 0-65535)
…	Zero Padding to complete the 55 bytes assigned for single event (if it's a logged event, i.e. sent actively)

3 Command Channel (Inbound Messages)

3.1 Overview

The command channel comprises seven types of messages, as described in the following sections:

- **Generic Command (Message Type 0)** – some commands are sent using this legacy message. This message is always replied with a legacy status/location message from the target unit (if the command is received successfully).
- **Programming Command (Message Type 1)** – this message provides OTA programming capabilities, and is always replied to with a programming data message from the target unit, when received correctly.
- **Generic Acknowledge Message (Message Type 4)** – this message is sent by the server to verify reception of outbound status/location, telemetry or transparent data messages.
- **Modular Message Request (Message Type 9)** – this legacy modular message is designed to request the unit to send types of data, defined in Modular Message packet like CAN bus sensors, Cell ID, debug data, etc.
- **Self Re-flash Chunks (Message Type 10)** – this message forwards firmware file data chunks for the self-re-flash process of the unit. The Self re-flash process description is outside of the current scope of this document.

3.2 Generic Command (Message Type 0)

The generic command message is the main command interface to the unit.

3.2.1 *Message Ingredients*

- Message header
  - System Code – 4 bytes
  - Message Type – 1 byte
  - Unit ID – 4 bytes
  - Command Numerator – 1 byte
  - Authentication Code – 4 bytes
- Command Code – 1 byte (repeated twice)
- 1st Command Data Field – 1 byte (repeated twice)
- 2nd Command Data Field – 1 byte (repeated twice)
- Command Specific Data Field – 4 bytes
- Error Detection Code – 1 byte

3.2.2 *Byte-Aligned Table*

Byte	Description
1	System Code, byte 1 – ASCII "M"
2	System Code, byte 2 – ASCII "C"
3	System Code, byte 3 – ASCII "G"
4	System Code, byte 4 – ASCII "P"
5	Message Type (0)
6	Unit ID
7	
8	
9	
10	Command Numerator
11	Authentication Code (Not Supported, currently sends as zeros)
12	
13	
14	

Byte	Description
15	Command Code
16	Command Code (repetition)
17	1st Command Data Field
18	1st Command Data Field (repetition)
19	2nd Command Data Field
20	2nd Command Data Field (repetition)
21	Command Specific Data Field
22	
23	
24	
25	Error Detection Code (8-bit additive checksum, excluding system code)

3.2.3 *Detailed Per-Field Specifications*

3.2.3.1 Command Numerator Field

This field should contain the number of the command. This number appears in the "Message numerator" field in the unit reply message, enabling the user to easily distinguish between acknowledged commands and un-acknowledged ones.

3.2.3.2 Authentication Code

Not supported, currently sends as zeros.

3.2.3.3 Command Code

As the generic command message is relevant for all kinds of commands, it is necessary to specify the actual command that is desired. Therefore, each different command assigns a unique command code, which is used in the command code field, to specify the command to be executed.

3.2.3.4 Command Data Fields (1st and 2nd)

The command data fields contain further information, which is needed by some of the commands.

3.2.3.5 Command Specific Data Field

The command specific data field contains additional information, which is needed by some of the commands.

The available commands and corresponding data fields are detailed below:

Command Code (Hex)	Description
0x00	Immediate status request
0x02	0x02: Reset
0x03	Output state change Data field should contain output change information, according to this table: Data field 1 value: function 04h / 14h: Blinkers Output 1 (off / on) 05h / 15h: Standard immobilizer Output 2 (off / on) Example: Activate Blinkers MCGP 00 ID ID ID ID 00 00 00 00 00 03 03 14 14 00 00 00 00 00 00 CS
0x04	0 – Disable active transmissions 1 – Enable active transmissions Command Specific Data field: don't care
0x05	Tracking control command (based on time events). Data field: zero to stop tracking, non-zero sets the resolution of time events and immediately implements it. Command Specific Data field: don't care
0x0D	Erase tracking Log from NVM memory Data field: don't care
0x0E	Reset GPS receiver Data field: Zero for standard reset (by On/Off pin) 1st = 0x5A 2nd =0xA5 For Factory GPS reset command. Note that the unit can (configurable) perform GPS reset automatically in the following cases: • Standard reset (by On/Off pin) on ignition off. • If the GPS is communicating but not navigating and MODE1=0, MODE2=16 for 10 minutes, the unit performs a factory GPS reset. • If the GPS is not communicating, or communicating but not navigating and MODE1≠0, MODE2≠16 for 15 minutes, the unit performs standard GPS reset. • If same condition as in item 3 remains true for the next 15 minutes the unit performs a factory GPS reset.

Command Code (Hex)	Description
0x12	Connect to server (from FW28 and up) 0 – Main server 1 – Secondary server (provisioning) 2 – Maintenance Server

3.3 Programming Command (Message Type 1)

The programming command message enables you to configure the unit.

NOTE: For configuration spaces larger than 4K (typically in Cello-IQ and Cello-CANiQ units) it is mandatory to use Type 11 programming command (modules 10, 11).

3.3.1 *Message Ingredients*

- Message header
  - System Code – 4 bytes
  - Message Type – 1 byte
  - Unit ID – 4 bytes
  - Command Numerator – 1 byte
  - Authentication Code – 4 bytes
- Block Code – 1 byte
- Programming Masking Bitmap – 2 bytes
- Block Data – 16 bytes
- Error Detection Code – 1 byte

3.3.2 *Byte-Aligned Table*

Byte	Description
1	System Code, byte 1 – ASCII "M"
2	System Code, byte 2 – ASCII "C"
3	System Code, byte 3 – ASCII "G"
4	System Code, byte 4 – ASCII "P"
5	Message Type (1)

<table>
<tr><th>Byte</th><th>Description</th></tr>
<tr><td>6</td><td rowspan="4">Unit ID</td></tr>
<tr><td>7</td></tr>
<tr><td>8</td></tr>
<tr><td>9</td></tr>
<tr><td>10</td><td>Command Numerator</td></tr>
<tr><td>11</td><td rowspan="4">Authentication Code (Not supported, currently sends as zeros)</td></tr>
<tr><td>12</td></tr>
<tr><td>13</td></tr>
<tr><td>14</td></tr>
<tr><td>15</td><td>Block Code</td></tr>
<tr><td>16</td><td rowspan="2">Programming Masking Bitmap
<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">Byte 17</td><td colspan="8">Byte 16</td></tr>
</table></td></tr>
<tr><td>17</td></tr>
<tr><td>18-33</td><td>Block Data</td></tr>
<tr><td>34</td><td>Error Detection Code (8-bit additive checksum, excluding system code)</td></tr>
</table>

3.3.3 *Detailed Per-Field Specifications*

3.3.3.1 System Code

Refer to Section 3.2.3.1

3.3.3.2 Message Type

Programming Command messages contain a value of 1 (one) in the message type field.

3.3.3.3 Unit ID

Refer to section 3.3.3.33.2.3.3

3.3.3.4 Command Numerator Field

Refer to section 3.2.3.4

3.3.3.5 Authentication Code

(Not Supported, currently sends as zeros)

Refer to section 3.2.3.5

3.3.3.6 Block Code

OTA (over the air) parameter programming is done in blocks. The entire NVM parameter memory is partitioned to 16-byte long blocks. Each of those blocks is uniquely identified with a block code. The block code field contains the code of the block whose data is sent in this message (in the block data field).

3.3.3.7 Programming Masking Bitmap

The bitmap allows programming of only part of the parameters in a block, while leaving the other parameters with their previous values.

Each bit in the 16-bit value represents a byte in the parameter memory block. A value of "1" in a certain bit enables programming to the corresponding byte in the parameters memory, where a value of "0" prohibits programming of that byte.

3.3.3.8 Block Data

Contains the actual data programmed in the specified block of the parameter memory.

3.4 Generic Acknowledge Message (Message Type 4)

The generic acknowledge message is an inbound message sent by the server to verify reception of outbound Status/Location (Type 0), Data Forwarding (Type 7, 8) and Modular (Type 9) messages.

3.4.1 *Message Ingredients*

- Message header
  - System Code – 4 bytes
  - Message Type – 1 byte
  - Unit ID – 4 bytes
  - Command Numerator – 1 byte
  - Authentication Code – 4 bytes (Not Supported, currently sends as zeros)
- Action Code – 1 byte
- Main Acknowledge Number – 2 bytes (1 reserved)
- Secondary Acknowledge Number – 2 bytes (reserved)
- Compressed Date – 2 bytes
- Compressed Time – 2 bytes
- Spare – 2 bytes
- Error Detection Code – 1 byte

3.4.2 *Byte-Aligned Table*

Byte	Description
1	System Code, byte 1 – ASCII “M”
2	System Code, byte 2 – ASCII “C”
3	System Code, byte 3 – ASCII “G”
4	System Code, byte 4 – ASCII “P”
5	Message Type (4)
6	Unit ID
7	
8	
9	
10	Command Numerator
11	Authentication Code (Not supported, currently sends as zeros)

<table>
<tr><th>Byte</th><th>Description</th></tr>
<tr><td>12</td><td rowspan="3"></td></tr>
<tr><td>13</td></tr>
<tr><td>14</td></tr>
<tr><td>15</td><td>Action Code (sent as zero)</td></tr>
<tr><td>16</td><td>Main Acknowledge Number – LSB</td></tr>
<tr><td>17</td><td>Reserved for Main Acknowledge Number – MSB (sent as zeros)</td></tr>
<tr><td>18</td><td>Reserved for Secondary Acknowledge Number – LSB (sent as zeros)</td></tr>
<tr><td>19</td><td>Reserved for Secondary Acknowledge Number – MSB (sent as zeros)</td></tr>
<tr><td>20</td><td>Reserved for future use (sent as zeros)</td></tr>
<tr><td>21</td><td rowspan="2">Compressed Date
<table>
<tr><td colspan="5">Day</td><td colspan="4">Month</td><td colspan="7">Year (-2000)</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">Byte 22</td><td colspan="8">Byte 21</td></tr>
</table></td></tr>
<tr><td>22</td></tr>
<tr><td>23</td><td rowspan="3">Compressed Time
<table>
<tr><td colspan="7">Spare (sent as 128)</td><td colspan="6">Seconds</td><td colspan="6">Minutes</td><td colspan="5">Hours</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">Byte 25</td><td colspan="8">Byte 24</td><td colspan="8">Byte 23</td></tr>
</table></td></tr>
<tr><td>24</td></tr>
<tr><td>25</td></tr>
<tr><td>26</td><td rowspan="2">Spare</td></tr>
<tr><td>27</td></tr>
<tr><td>28</td><td>Error Detection Code (8-bit additive checksum, excluding system code)</td></tr>
</table>

3.4.3 *Detailed Per-Field Specifications*

3.4.3.1 Action Code

Sent as zero.

3.4.3.2 Main Acknowledge Number

This field contains the Message Numerator filed of the acknowledged outbound message.

3.4.3.3 Secondary Acknowledge Number

Currently not used and sent as zero.

3.5 Modular Message Request (Message Type 9)

The modular data packet request is designed to provide different data types in the same packet.

3.5.1 *Message Ingredients*

- Message header
  - System Code – 4 bytes
  - Message Type – 1 byte
  - Unit ID – 4 bytes
  - Command Numerator – 1 byte
- Authentication Code – 4 bytes (Not supported, currently sends as zeros)
- Packet Control Field – 1 byte
- Message Length – 1 byte
- First Sub-Data Type – 1 byte
- First Sub-Data Length – 1 byte
- First Sub-Data variable length, depends on Data Type
- ……
- Nth Sub-Data Type – 1 byte
- Nth Sub-Data Length – 1 byte
- Nth Sub-Data– variable length, depends on Data Type N
- Error Detection Code – 1 byte

3.5.2 *Byte-Aligned Table*

Byte	Description
1	System Code, byte 1 – ASCII "M"
2	System Code, byte 2 – ASCII "C"
3	System Code, byte 3 – ASCII "G"
4	System Code, byte 4 – ASCII "P"
5	Message Type (9)
6	Unit ID (total 32 bits)
7	
8	

Byte	Description
9	
10	Command Numerator
11	Authentication Code (Not supported, currently sends as zeros)
12	
13	
14	
15	Packet Control Field
16	Length (of the modules section - not including the checksum)
17	First Sub-data Type
18	First Sub-data Length
19	First Sub-data Data
…	…
	Nth Sub-data Type
	Nth Sub-data Length
	Nth Sub-data Data
Last Byte	Error Detection Code (8-bit additive checksum, excluding system code)

3.5.3 *Detailed Per-Field Specifications*

3.5.3.1 Packet Control Field

Direction	Out of space indication	Unused					
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Direction

0 – Data from the unit

1 – Request (unit-bound)

Out of Space Indication

0 – All the requested data is present in the message.

1 – Some Sub-data was not returned due to data size.

3.5.3.2 Length

This field includes the number of data bytes of the modules (with their types and lengths). It is the number of bytes from byte 17 to the byte of the checksum, which is not included.

3.5.4 *Inbound Sub-Data Types Table*

Code (Hex)	Function
0x01	Firmware Platform Manifest Request
0x08	Authentication Table Update Command (Not supported, currently sends as zeros)
0x09	Cell ID Request
0x12	Modular Platform Manifest Request

3.5.5 *Firmware Platform Manifest Request*

This sub-data serves as a Firmware Manifest Request. The unit responds to this sub-data with Firmware Platform Manifest sub-data (0x01).

Byte	Description
0	Sub-Data Type (0x01)
1	Length – 0

3.5.6 *Authentication Table Update Command*

Not Supported, currently sends as zeros.

3.5.7 *Cell ID Request*

This sub-data causes the unit to generate a Type-9 outbound message, containing the last known Cell ID related information (updated every 60 seconds). The generated message will vary according to unit modem: 2G – sub-data 0x09, 3G – 0x0C, CDMA – 0x1C, 4G – 0x1D.

Byte	Description
0	Sub-Data Type (0x09)
1	Length – 2
2	Cell ID Request (0x09)
3	Spare

3.5.8 *Modular Platform Manifest Request*

This command causes the unit to generate an OTA Modular Platform Manifest message. The message will contain the data fields as per the specification in the command.

Data part: The data part of this packet has a size of 6 bytes. Each byte contains a bitmask as described below. Setting a bit to "1" causes the unit to add a corresponding field to the Modular Platform Manifest.

Byte	Description
0	Sub-Data Type (0x12)
1	Length – 6
2	Bit Map 0
3	Bit Map 1

Bit Map 0 (Byte 2):

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Size of external RAM	Amount of ext. non-volatile memory used by application (e.g. configuration)	Size of external non-volatile memory	Size of internal RAM	Amount of non-volatile memory used by application (e.g. configuration)	Size of Program memory	Accelerometer identifier	Processor identifier

Bit Map 1 (Byte 3):

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FW Upgrade Date/Time	First Activation Date/Time	GPS firmware	GPS Type	Modem firmware	Modem type	Current Hardware ID number	Current Firmware ID number

Byte	Description							
4	Bit Map 2							
	Originating FW ID	IMSI/IMEI/MEID	Cello-CANiQ VIN (Vehicle ID)	DFD/SD card version	Boot loader ID	System ID (STM ID in case of STM controller)	Firmware name (string)	Last Configuration Change Date/Time
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
5	Bit Map 3							
	Maintenance Configuration	Modem Firmware Sub Version	PIN#8 HW Selected Function	SIM ICCID	BT Extender Identification	Size of internal non-volatile memory request	Version Information	Nano sensors
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
6	Bit Map 4							
	Reserved	Reserved	Reserved	Reserved	Reserved	Battery Fuel Gauge Identification	Silicon Labs 4632 receiver version	BT version number
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
7	Bit Map 5							
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Note: Reserved bits will be sent as 0.