Cellocator Serial Communication Protocol

Proprietary and Confidential

Version 6.1.3.0

Revised and Updated: Sept 13, 2021

CELLOCATOR - A POWERFLEET® BRAND. 14 HAMELACHA ST. ROSH HA'AYIN 4809133, ISRAEL • TEL: 972-3-5723111 • WWW.CELLOCATOR.COM

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 PowerFleet Inc. 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 PowerFleet Inc. Using, integrating, and/or selling the product for use in such applications do so at their own risk and agree to fully indemnify PowerFleet Inc. For any resulting loss or damages.

Warranty Exceptions and Disclaimers

PowerFleet Inc. 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 PowerFleet Inc. 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 PowerFleet Inc., 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 PowerFleet Inc., the purchaser shall be solely responsible for the proper configuration, testing and verification of the product prior to deployment in the field.

POWERFLEET INC.'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. POWERFLEET INC. 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 POWERFLEET INC. 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 POWERFLEET INC. 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 direct wire interface specifications for interconnection between Cellocator devices and external units. Included in this document is all relevant information about the structure of outbound and inbound packets.

1.2 FW Version Applicability Table

This document describes the serial protocol of both the Legacy fleet system and the new Cellocator Safety Application (CSA).

Unit	FW Version supported (From)
Cello-F	31x
Cello-R	31x
CelloTrack	65a
CelloTrack-Nano	34a
CR200	31z
CR200B	31z
Cello-IQ40, Cello-IQ50	32e, 32h
Cello-IQ40, Cello-IQ50 GNSS	33e
Cello-CANiQ	33e
CR300	41q
CelloTrack-4 family	65a
Cello-4	81a
CR400	54a

1.3 Revision History

Version	Date	Description
28a	7/5/09	Original version
28a 1.1	9/7/09	Added Platform Manifest packet description
28f 1.2	17/1/10	Fixed command numbers of the "Old" Platform Manifest packet Added Modular Platform Manifest packet description
30.0b	28/3/10	Adaptation to the CelloEDGE
30.01	7/4/10	Released
30.02	14/4/10	Added to shipment mode description **NOTE:** The shipment mode is activated by default. It means, that after complete power loss (both main and backup power supplies are disconnected) the battery will be disconnected (and reconnect only upon reconnection of the main power).
31.01	18/5/10	Authentication table read/write command and response to it fixed.
31c.01	28/5/11	Added new data forwarding type Extended ACK (type 0x02)
31c.02	5/12/11	Fixed description of Master Unit Status Message (Type 0x08), which is transmitted as a reply to a Status Request Command (type 10h)
31h.01	22/5/12	Added CelloTrack output reference, fixed inputs of CelloTrack
31j.01	22/5/12	Voice Call dialing inbound command and outbound Nack, and response.
31m.1	19/11/12	Serial Upload Mode Control and Status (0x2c) (Added as infrastructure for CFE harvester)
31m.1	16/12/12	Add support for Cellotrack3G IICID query and response.
31q	16/12/12	Add support for Cellotrack3G IICID query and response.
V6.0.0.2	24/12/12	Add Cello-IQ serial protocol. Cello: 31n.1 Cello-IQ: 32a

Version	Date	Description
31u	06/03/13	Updates for Cellotrack3G: VIN – Vehicle Identification Number : Add header and mark as (Infrastructure)
31s	20/03/13	Update version number
31x	20/05/13	Update version number
V6.0.0.3 (32b)	27/1/13 , 21/5/13	Add Cello-IQ serial protocol. Updated modem versions
V6.0.0.4	9/10/13	Updated modem versions: Same as wire protocol
V6.0.0.6	29/10/13	Section 4.19.1: Modem Revision ID: Updated new Telit modems firmware version IDs
V6.0.0.7	05/10/13	Section 4.9: Platform Manifest (old Type) The detailed fields data was removed and replaced with reference to "Cellocator Wireless Communication Protocol" Document. Section 4.19.1: The section was removed and replaced with reference to the "Cellocator wireless communication protocol" Document
V6.0.0.8	05/11/13	General: All messages now contain M2C header and the byte numbers are reordered to start from the first byte ('M'). Section 6.29.14: remove the languages table and replace with a reference to CSA Programming manual.
V.6.0.0.9	30/12/2013	Section 4.6: Remove comment which was relevant only for Compact units
V6.0.0.10	20/1/14	Update year to 2014
V6.0.0.11	10/4/14	The document section hierarchy was restructured. Section 1.1: added Fw Version Applicability table
V6.0.0.12	02/6/14	Restructure the command summary tables
V6.0.0.13	30/6/14	Section 3.3.14: Unit I was missing in the Shipment Mode Status message.
V6.0.1.0	15/1/15	Reformatted by technical writer.
6.0.1.1	25/3/15	Section 3.3.8: changed reference to wireless protocol: 2.6.1 → 2.7.3 Section 3.3.17: changed reference to wireless protocol: 2.6.16 → 2.7.16
6.0.1.2	23/6/15	Section 3.3.2.1 (3.3.3): promoted chapter numbering of type 0x02 message from 3.3.2.1 to 3.3.3

Version	Date	Description
6.0.1.3	29/6/15	Section 4.4.2: • Fixed message type in the table from 0x00 to 0x01 • Deleted data bytes enumerations in the "command" column in data bytes description table
6.0.1.4	7/7/15	Section 3.3.9: Fixed "Size of program memory" (bytes 11-12) to 1024 bytes units instead of 1024 words units
6.0.1.5	4/8/15	Section 4.4.27: Fixed the NACK response from type 01 to type 02 Section 3.3.3.4: Fixed "Communication Control Byte" to "Unit's status + Current GSM Operator Byte" Section 3.3.3.7: • Fixed "Communication Control Byte" to "Unit's status + Current GSM Operator Byte" • Deleted irrelevant description of Home/Roam flag Section 3.3.6: Added byte aligned table Section 3.3.3.1: Added elaboration for escorting data
6.0.1.6	10/8/15	Section 3.3.3: In "Available space in buffer" field (bytes 13-14), fixed buffer size from 1024 to 1020 bytes Section 3.3.3.1: • Fixed buffer size from 1024 to 1020 bytes • Added optional FM 48 bytes to escorting bytes
6.0.7.0	16/8/15	Section 3.3.19: Fixed referenced Modular Platform Manifest in wireless protocol section number from 2.6.16 to 2.7.17

Version	Date	Description
6.1.0.0	16/3/16	Section 3.2: Added PointerCept base station outbound messages Added section (3.3.21) for PointerCept Base Station – Beacon Detected message (040h) Added section (3.3.22) for PointerCept Base Station – Status message (041h) Added section (3.3.23) for PointerCept Base Station – Power Status message (042h) Added section (3.3.24) for PointerCept Base Station – Location Request message (043h) Added section (3.3.25) for PointerCept Base Station – User Registration/Connection Approval message (044h) Section 4.3: Added PointerCept base station inbound messages Section 4.4.26: • Added Bit Map Bytes 4, 5 of Modular Platform Manifest Request • Added PointerCept base station BT version number and Silicon Labs 4632 receiver version (Bit Map Byte 4, bits 0 and 1, respectively) Added section (4.4.35) for PointerCept Base Station – Status Request/Power Control message (040h) Added section (4.4.36) for PointerCept Base Station – Chase Start/End/Location Update message (041h)
6.1.0.1	9/6/16	Section 3.3.22: • Changed CPU Temperature (Status message (0x41), byte 21) to NTC Temperature • Fixed Vin resolution (Status message (0x41), byte 18) from 0.1466V/bit to 0.117647V/bit
6.1.1.0	9/8/16	Section 3.3.19: Replaced section number with module name as reference in Wireless Protocol document Section 4.4.26: • Added Version Information (Cello-CAN(IQ)) to bit map byte 3, bit 1 • Added Size of Internal Non-Volatile Memory to bit map byte 3, bit 2 • Added BT Extender Identification to bit map byte 3, bit 3
6.1.1.1	9/11/16	Section 3.3.8.4: Added enum 20 (Packet connection by TLS socket) to cellular status report in Master Unit Status Message (0x08)

Version	Date	Description
6.1.1.2	21/12/16	Section 4.4.26: Added SIM ICCID in Bit Map 3 (Byte 8), bit 4
6.1.1.3	12/01/17	Section 3.3.3: Changed "Movement/Idle" (message type 0x02, byte 15, bit 4) to "Driving Status" and added elaboration for 0/1 values Section 3.3.3.6: Changed "Movement/Idle" to "Driving Status" and added elaboration to match the wireless protocol Section 3.3.21: Changed "PointerCept Transmission Reason" (message type 0x40, byte 33) to "Transmission Reason" and added elaboration regarding this field
6.1.1.4	30/01/17	Section 4.4.26: Added in 3rd byte of the bitmap, in bit 5, the "PIN#8 selected function"
6.1.1.5	23/02/17	Section 4.4.37: Added the Nano status interrogation command.

Version	Date	Description
6.1.2.0	29/05/17	Section 3.2: • Added Authenticated Features Query Response (Type 0x09) • Added Modem FOTA Response (0x50) • Deleted Features Matrix Query and Response (0x2E) Section 3.3.2: • Added NACK codes 0x46, 0x47 for authenticated features • Added NACK codes 0x5A, 0x5B, 0x5C, 0x5D, 0x5E for Modem FOTA process Section 3.3.8.4: Added Cellular Statuses 30, 31 for Modem FOTA process Added Section 3.3.9 for Authenticated Features Query Response Added section 3.3.26 for Modem FOTA Response Section 4.3: • Added Authenticated Features Command (Type 0x09) • Added Modem FOTA Command (Type 0x50) • Added Nano Status Interrogation (Type 0x37) Section 4.4.27: • Added Modem Firmware Sub Version in Bit Map 3 (Byte 8), bit 6 • Added Maintenance Configuration in Bit Map 3 (Byte 8), bit 7 Added section 4.4.7 for Authenticated Features Command Added section 4.4.39 for Modem FOTA Command
6.1.2.1	4/1/18	Section 4.3: Added 2 new commands for Nano (FW 34j) 0x39 and 0x3A. Section 4.4.34: Details on the "Request to send a distress event to server" command.
6.1.2.2	18/4/18	Section 3.3.8.1 and 3.3.8.8: Added in byte 52 (for Nano only) a Server ping status bits. Section 4.4.35: Details on the "Push RTC" command.
6.1.2.3	23/5/18	Section 4.4.13.2: Changed Red/Green LED of the AI100 to be in Byte 10 bits 5-6, and leave Byte 11 bit 5 to be Standard immobilizer.
6.1.2.4	7/11/18	Sections 3.3.8.1, 4.4.34, 4.4.35: Added that also the CelloTrack-4 family support these commands.

Version	Date	Description
6.1.2.5	28/5/19	Section 3.2: Added opcode 70: Tunneling directly to modem
6.1.2.6	5/8/19	Sections 4.3 and 4.4.36: Added opcode 71 - Unit inactivation
6.1.2.7	16/2/20	Section 4.3: Added Packets Reservations for CelloTrack-Phoenix (LV300P), of packets: 0x48 and 0x49. Section 3.3.2: Added data part Value "F2", per error message generation "Error: Cannot operate Phoenix interface when "CelloTrack GPIO Control 2" is Enabled! Please Disable it and try again." Section 1.2: Added Applicability for CelloTrack and CelloTrack-Nano.
6.1.2.8	26/2/20	Section 3.3.26: Added packet for "Data fields over serial".
6.1.2.9	14/10/20	Section 3.3.8.4: Added more cellular status codes for CR400 combined modem and Satellite modem of CT-Solar V3. Sections 4.3 and 4.4.37: Added new inbound command for Pass through (0x46).
6.1.2.10	23/11/20	Section 3.3.4: Rewritten the message structure of the type 0x04 outbound, with the correct content. Section 4.4.13: Improved readability and bit map more accurate now. Section 4.4.37: Added new device ENUM (4-BT) for Pass through (0x46) command.
6.1.2.11	10/3/21	Section 1.2: Updated applicability table. Section 3.3.8.1: Added "Buses activity status" bytes in the Master unit status for Cello-4.
6.1.3.0	13/9/21	Added support of CANiQ-M (Master unit status)

2 Communication Layer Description

Direct wired communication between Cellocator devices and external units is achieved through a 3-wire connection (or 4-wire in older units). The voltage levels and data format are compliant with the RS-232 standard, which allows for a direct connection to a PC, without any adapters.

There are two available interface types:

- The 3-wire connection carries the following signals: serial data output line from the unit (DCE-RXD), serial data input to the unit (DCE-TXD) and common signal ground.
- The 4-wire connection (obsolete) carries the following signals: serial data output line from the unit (DCE-RXD), serial data input to the unit (DCE-TXD), handshake input line to the unit (DCE-RTS) and common signal ground.

Communication is packet-oriented. The unit generally does not initiate transmission of packets to the attached terminal (except for NMEA data forwarding from GPS, Garmin and PSP interfaces), unless data are received in the wireless channel, in which case the data are immediately forwarded to the attached terminal. The terminal must therefore always be prepared to receive incoming messages. Transmitting a packet to the unit, on the other hand, requires handshaking.

Further details are supplied in the appropriate part of this document. Data are transmitted on the DCE-RXD line in standard UART/NRZ serial asynchronous format at 9600bps (in CelloEDGE different baud rates are available as well), with 8 data bits, one start bit, one stop bit, and no parity bits. Received data are expected at the DCE-TXD line with the same format as the transmitted data.

All communication is performed in a little-endian manner (Intel-style), which means least significant bits are sent first. In multiple-byte fields, the least significant bytes are sent first.

3 Outbound Packets (Sent from Unit)

3.1 Overview

As explained in Section 2, outbound packets are sent without handshaking. All outbound packets follow the same basic structure, as described here:

1. Header (3 bytes, ASCII "M2C").
2. Packet type code (see types below, 1 byte).
3. Unit's serial number (4 bytes).
4. Packet data (number of bytes dependent of packet type)
5. Checksum (1 byte).

All packets contain parts 1, 2, 3 and 5. The only part that varies among the different packet types is part 4 (packet data) whose length is a function of the packet type, and possibly even the packet data itself. Some packet types do not contain a data part at all.

The checksum field is the additive unsigned sum, modulo 8-bit, of all bytes in the packet, excluding the header part and the checksum part itself. It can be expressed mathematically as the following:

$$Checksum = \mathrm{mod}\, 2^8 \sum_{i=4}^{8+N} Byte(i)$$

Where *N* is the length of data part (which is zero, in the absence of a data part).

3.2 Packet Types Summary

The following packet types are hereby defined:

Type Code (Hex)	Type Code (Dec)	Packet Type
00	00	Generic acknowledge packet.
01	01	Generic non-acknowledge packet (reporting error)
02	02	Unit status/information packet
03	03	Reserved for manufacturer use (debug purposes)
04	04	Unit inputs status packet
05	05	16 bytes of manufacturing traceability data
06	06	Data forwarding from wireless channel
07	07	GPS packet
08	08	Master unit status message

Type Code (Hex)	Type Code (Dec)	Packet Type
09	09	Authenticated Features Query Response
0B	11	EEPROM contents dump packet
0C	12	Forwarding modem's response to AT command
0E	14	Voice call status
0D	13	Forwarded CAN frame
15	21	Usage counters update
16	22	Command authentication table content packet
18	24	Shipment mode status report
19	25	Debug log
1A	26	Reserved for manufacturer use (debug purposes)
1B	27	
1C	28	Firmware image info
1D	29	Modular Platform manifest
20-22	32-34	Reserved for manufacturer use (debug purposes)
28	40	IICID Query and response
29-2B	41-43	Reserved
2C	44	Cello – Serial Upload Control Outbound and Inbound
2D	45	Serial Upload Process Outbound and inbound
31	49	Reserved: OP Session Description query
32	50	Cello-IQ and Cello-CANiQ DFD
33	51	Reserved (CFE)
35	53	Voice Call Management
37	55	Nano status interrogation
38	56	Reserved: Nano PT
3C	60	Data fields over serial
40	64	PointerCept Base Station – Beacon Detected
41	65	PointerCept Base Station – Status
42	66	PointerCept Base Station – Power Status

Type Code (Hex)	Type Code (Dec)	Packet Type
43	67	PointerCept Base Station – Location Request
44	68	PointerCept Base Station – User Registration/Connection Approval
50	80	Modem FOTA Response

The actual structure of each of these packet types is defined later in this document.

3.3 Detailed Outbound Packets Structures

3.3.1 *Generic Acknowledge Packet (Type 0x00)*

The generic acknowledge packet is sent after the successful processing of any command that does not have a specific response packet (e.g. write EEPROM command).

In a **Forward data to wireless channel** command, a Generic Acknowledge Packet is sent only if the unit is registered in a Home or Roaming cellular network.

Data part: the packet does not contain a data part.

3.3.2 *Generic Non-Acknowledge Packet (Type 0x01)*

The generic non-acknowledge packet is sent after unsuccessful processing of a command, or may be sent as a response to an illegal packet that was received.

Data part: The data part of this packet is 1 byte long.

Value (hex)	Description	
00	The command processing failed as a result of syntax problem (as illegal packet number etc.). * Message with checksum error will not cause NACK. It will be ignored by the unit.	
01*	The command was not forwarded to a wireless channel as the unit is not registered in a cellular network. * Applicable only with Forward data to wireless channel command.	
02	Upload commands received but upload mode is not established.	Self-reflash process. Nack
03	Upload commands received after completion was already performed.	Self-reflash process. Nack
06	Faulty phone number.	NAKs, defined for Voice call management.
07	Dial requested, but voice is already active.	
08	Voice call prohibited, or currently unavailable.	
09	Got a command for an active voice state, but voice is inactive.	
10	Incompatible script / re-flashing version.	Self-reflash process. Nack
20	Establishment failed due to incompatible platform.	Self-reflash process. Nack
40	CRC-32 test failed.	Self-reflash process. Nack

Value (hex)	Description	
46	Exceeded Number of Failed Feature Authentication Attempts (the unit will ignore Feature Authentication command for the next hour)	Authenticated Features NACK
47	Feature Authentication Code Discrepancy	Authenticated Features NACK
5A	Modem FOTA Process Cannot Start due to Operation on Internal Battery	Modem FOTA Process
5B	Modem FOTA Process with Maintenance Server Cannot Start because Disabled in PL	
5C	Modem FOTA Process Cannot Start due to Ongoing FW Upgrade Process	
5D	Modem FOTA Process Cannot Start due to Ongoing PL Upgrade Process	
5E	Modem FOTA Process Cannot Start due to Ongoing Modem FOTA Upgrade Process	
E0	External EEPROM verifies failure – either after chunk save command or after trying to save some parameters.	Self-reflash process. Nack
F0	Internal EEPROM programming error (auto-verification failure).	
F1	Busy, try again later.	
F2	Error: Cannot operate Phoenix interface when "CelloTrack GPIO Control 2" is Enabled! Please Disable it and try again.	

3.3.3 *Unit Status/Information Packet (Type 0x02)*

The unit status packet is sent as a response to a received status request packet.

It carries the current status of the unit, as well as general informative data.

Data part: The data part of this packet is 8 bytes long as listed in the table below.

Byte	Data
1	'M'
2	'2'
3	'C'
4	0x02
5	Unit ID
6	
7	

<table>
<tr><th>Byte</th><th>Data</th></tr>
<tr><td>8</td><td></td></tr>
<tr><td>9</td><td>Hardware Version</td></tr>
<tr><td>10</td><td>Software Version</td></tr>
<tr><td>11</td><td>Protocol Version</td></tr>
<tr><td>12</td><td>Software Subversion</td></tr>
<tr><td>13</td><td rowspan="2">Available space in buffer for data forwarding (bytes, Little endian, valid range 0-1020, from v31c)</td></tr>
<tr><td>14</td></tr>
<tr><td>15</td><td>
<table>
<tr><td>Unused</td><td>ACK/NACK

0 – ACK
1 – NACK</td><td>GPS valid / Invalid</td><td>Driving Status</td><td>GPRS Socket available</td><td>Home / Roam
0 – Home
1 – Roam</td><td>SMS available</td><td>Registered to GSM</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>
from v31c
</td></tr>
<tr><td>16</td><td>Spare</td></tr>
<tr><td>17</td><td>Checksum</td></tr>
</table>

3.3.3.1 Available Space in Buffer for Data Forwarding

2 bytes field, containing the amount of free bytes in a data RAM forwarding buffer. Little endian; valid range 0-1020.

This field will monitor the remaining space after attachment of escorting data for each incoming packet (forwarded message code byte + fragment control byte + 2 length bytes + optional 48 bytes of FM data – if "Forward Data as Container" is enabled).

3.3.3.2 Registered to GSM

Changes to '1' upon registration to GSM network. The bit is changing its status by first response to CREG command.

3.3.3.3 SMS Available

Changes to 1 when:

- GSM is available
- SMS in current GSM network and hibernation mode is enabled

3.3.3.4 Home (0)/ Roaming (1)

Same as in a corresponding bit in Unit's status + Current GSM Operator Byte of OTA msg. type 0.

3.3.3.5 GPRS Socket Available

Changes to 1 upon successful establishment of GPRS socket, changes back upon loss of GPRS socket.

3.3.3.6 Driving Status

This bit provides indication if the unit is in logical Ignition On/Off, according to the configuration of the detection source (physical ignition or accelerometer). It will indicate "1" when logical Ignition On is detected, and "0" when logical Ignition Off is detected. It corresponds to the bit in OTA message Type 0, 1st input byte, bit 5.

3.3.3.7 GPS Valid / Invalid

Same as in a corresponding bit in Unit's status + Current GSM Operator Byte of OTA msg. type 0.

3.3.4 *Unit Inputs Status Packet (Type 0x04)*

This packet is sent as a response to a unit input request packet. It carries the current status of all unit inputs.

Byte	Data
1	'M'
2	'2'
3	'C'
4	Message type - 0x04
5	Unit ID
6	
7	
8	
9	Inputs bitmap (Same as 1st and 2nd byte of IOs in wireless protocol)
10	
11	Spare = 0x0000
12	
13	Analog mapped by PL "Byte 26 of Msg 0"
14	Analog mapped by PL "Byte 27 of Msg 0"
15	Analog mapped by PL "Byte 28 of Msg 0"
16	Analog mapped by PL "Byte 29 of Msg 0"
17	CheckSum

3.3.5 *16 bytes of Manufacturing Traceability Data (Type 0x05)*

This packet is sent as a response to a manufacturing traceability data packet request (Type 5). It carries the current status of the field.

Data part: The data part of this packet is 16 bytes long.

Header			Type	Block data (16 bytes)																CS
M	2	C	05																	

3.3.6 *Data Forwarding from Wireless Channel (Type 0x06)*

This packet is sent unsolicited, after an appropriate message is received at the wireless channel. Once such a message arrives, the data are immediately wrapped in this packet and forwarded to the attached terminal.

Data part: First byte is the number of bytes that are being forwarded. The rest of the bytes (at least one) are the actual forwarded data.

Byte	Data
1	'M'
2	'2'
3	'C'
4	Message type – 0x06
5	Payload length (Valid values: 1-255)
6 to 6+X	The payload (X bytes)
7+X	Check Sum

3.3.7 *GPS Data (Type 0x07)*

This packet is sent as a response to a GPS Data request packet. It carries the current data from the GPS receiver.

3.3.7.1 Data Part (Constant Size 28 Bytes)

Byte number	Description
1	'M'
2	'2'
3	'C'
4	Message type – 0x07
5	Mode 1 (from GPS)
6	Mode 2 (from GPS)
7	Number of satellites used (from GPS)
8	Longitude
9	

Byte number	Description
10	
11	
12	Latitude
13	
14	
15	
16	Altitude
17	
18	
19	
20	Ground speed
21	
22	
23	
24	Speed direction (true course)
25	
26	UTC time – seconds
27	UTC time – minutes
28	UTC time – hours
29	UTC date – day
30	UTC date – month
31	UTC date – year
32	CheckSum

3.3.7.2 MODE 1 From GPS

This field is generated by the GPS. It is a bitmapped field of flags, defined in the following manner:

Table A-39 Mode 1

Bit	7	6	5	4	3	2	1	0
Bit(s) Name	DGPS	DOP-Mask	ALTMODE		TPMODE	PMODE		

Bit(s) Name	Name	Value	Description
PMODE	Position mode	0	No navigation solution
		1	1 satellite solution
		2	2 satellite solution
		3	3 satellite solution
		4	>3 satellite solution
		5	2D point solution (Least square)
		6	3D point solution (Least square)
		7	Dead reckoning
TPMODE	Trickle power mode	0	Full power position
		1	Trickle power position
ALTMODE	Altitude mode	0	No altitude hold
		1	Altitude used from filter
		2	Altitude used from user
		3	Forced altitude (from user)
DOPMASK	DOP mask status	0	DOP mask not exceeded
		1	DOP mask exceeded
DGPS	DGPS status	0	No DGPS position
		1	DGPS position

With **Normal PMODE Filter** settings (In EEPROM of the unit), the unit considers a valid fix as a fix which has PMODE: 2 to 6 (0, 1 and 7 are considered invalid).

With **Tide PMODE Filter** settings (In EEPROM of the unit), the unit considers a valid fix as a fix which has PMODE: 3 and 4.

If these terms are not satisfied, a historic value is used instead.

3.3.7.3 MODE 2 from GPS

This field is generated by the GPS. It is a hexadecimal value, defined in the following manner:

Mode 2		Description
Hex	ASCII	
0 x 00	0	Solution not validated
0 x 01	1	DR Sensor Data
0 x 02	2	Validated (1)[1], Unvalidated (0)
0 x 04	4	If set, Dead Reckoning (Time Out)
0 x 08	8	If set, output edited by UI (i.e., DOP Mask exceeded)
0 x 10	16	Reserved
0 x 20	32	Reserved
0 x 40	64	Reserved
0 x 80	128	Reserved

3.3.7.4 Number of Satellites Used

Number of satellite measurements used for current position fix. Possible values: 0 to 12.

3.3.7.5 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 $-\pi$ to $+\pi$ for longitude, or $-\pi/2$ to $+\pi/2$ for latitude. The coordinates refer to WGS-84 map datum and ellipsoid.

3.3.7.6 Altitude

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

3.3.7.7 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).

3.3.7.8 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.

3.3.7.9 UTC Time

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

3.3.7.10 UTC Date

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

3.3.8 *Master Unit Status Message (Type 0x08)*

Master Unit Status Message may be transmitted as a reply to a request (type 0x010) or as an unsolicited message if the unit is set to send when updated or periodically. (Currently the message sends by request only).

3.3.8.1 Packet Structure

Byte	Data
1	'M'
2	'2'
3	'C'
4	0x08
5	Unit ID
6	
7	
8	
9	Spare
10	Peripheral flags
11	Hardware version
12	Software version
13	Software sub-version
14	Protocol version
15	Unit status – same as byte 16 in wireless protocol
16	Configuration flags – (0x0000)
17	
18	Unit mode – same as byte 20 in wireless protocol
19	I/O Status – same as bytes 21 –25 in wireless protocol
20	
21	
22	
23	

Byte	Data
24	Analog Input 1 – same as byte 26 in wireless protocol
25	Analog Input 2 – same as byte 27 in wireless protocol
26	Analog Input 3 – same as byte 28 in wireless protocol
27	Analog Input 4 same as byte 29 in wireless protocol
28	Mileage counter – same as bytes 30-32 in wireless protocol
29	
30	
31	Cellular management engine status
32	
33	PPP, IP protocol status
34	Unit's IP address (if applicable)
35	
36	
37	
38	Cellular registration status
39	Cellular location area code (LAC/TAC)
40	
41	Cellular Cell ID (Lower 2 bytes only)
42	
43	Cellular Network Code (=MCC*1000+MNC)
44	
45	
46	Last Detected Dallas ID
47	
48	
49	
50	
51	
52	Server ping status (in Nano and CelloTrack-4 only)
53	HDOP

Byte	Data
54	Buses activity status 1 (in CANiQ-M only, otherwise spare): Bit 0: CAN #1 Connected=1 / Disconnected (or disabled)=0 Bit 1: Spare Bits 2-5: CAN #1 bus rate (see ENUM below) Bit 6: K-Line Connected=1 / Disconnected (or disabled)=0 Bit 7: Spare
55	Buses activity status 2 (in Cello-4 only, otherwise spare): Bit 0: CAN #2 Connected=1 / Disconnected (or disabled)=0 Bit 1: Spare Bits 2-5: CAN #2 bus rate (see ENUM below) Bit 6: J1708 Connected=1 / Disconnected (or disabled)=0 Bit 7: Spare
56	Checksum

3.3.8.2 Data Specification

Peripheral flags indicate a problem with peripherals, e.g. no communication with cellular, no communication with EEPROM, etc.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Currently Unused							Modem Communication Error

3.3.8.3 Cellular Management Engine Status

This field contains the status of the cellular management software layer, indicating the state of the cellular interface from the unit's point of view.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Low	Cellular Status							
High	Currently Unused							

3.3.8.4 Cellular Status

0 – Cellular modem turned off

1 – Cellular modem turned on

2 – Cellular modem initialization

3 – SIM card PIN code checking

4 – Network registration checking
5 – Network information reading
6 – Incoming SMS checking
7 – Incoming SMS reading
8 – Incoming SMS deleting
9 – Outgoing SMS sending
10 – Cellular idle
11 – Outgoing CSD dialing
12 – Packet connection dialing
13 – Connection closing
14 – Voice connection
15 – Packet connection by CSD
16 – Packet connection by UDP
17 – Packet connection by TCP
18 – Back to packet connection
19 – Processing AT command from serial port
20 – Packet connection by TLS socket
30 – Modem FOTA Upgrade File Download
31 – Modem FOTA Upgrade Process
32 – GNSS on/off query (CR400 ME910 GNSS and Modem common module)
33 – GNSS turning ON (CR400 ME910 GNSS and Modem common module)
34 – GNSS turning OFF (CR400 ME910 GNSS and Modem common module)
35 - Satellite Modem is turned OFF
36 - Satellite Modem is turned ON
37 - Satellite Modem Initializing
38 - Satellite Modem SBD Connection
39 - Satellite Modem Network Registration
40 - Satellite Modem Network Getting Info
41 - Satellite Modem Connected-Idle
42 - Satellite Modem in the power-down process
43 - Satellite Modem RSSI too low
44 - Satellite Modem Delay after error
45-254 - Reserved
255 – Illegal status value (for debugging use)

3.3.8.5 Cellular Registration Status

Result of +CREG command for registration status (not registered, home, roaming, etc). After a reset the value is 255 and changed to registration value (0, 1, 2, 3, 4 or 5) after first +CREG command.

3.3.8.6 PPP, IP Protocol Status

This field contains the status of the PPP and IP protocols (closed, opening, up, etc). The field currently is unused.

3.3.8.7 Unused Fields

The following fields are currently unused:

- Cellular Location Area Code
- Cellular ID
- Cellular Network Code

3.3.8.8 Server ping status (Nano only)

Server ping status byte:

Spare	Spare	Spare	Spare	Spare	Spare	Operational server returned ACK to last message 0- No 1- Yes	This feature is supported by this FW 0- No 1- Yes
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

3.3.8.9 CAN bus rate ENUM

Code	CAN Bus rate
0	125 Kbps
1	250 Kbps
2	500 Kbps
3	1 M Kbps
4	50 Kbps
5	62.5 Kbps
6	83.333 Kbps
7	100 Kbps
8	33.333 Kbps

9-14	Reserved
15	Auto-detect / Unknown

3.3.9 *Authenticated Features Query Response (Type 0x09)*

This module enables sending features codes upon receiving Authenticated Features Query Command (message 0x09).

<table>
<tr><th>Byte</th><th>Data</th></tr>
<tr><td>1</td><td>'M'</td></tr>
<tr><td>2</td><td>'2'</td></tr>
<tr><td>3</td><td>'C'</td></tr>
<tr><td>4</td><td>0x09</td></tr>
<tr><td>5</td><td rowspan="4">Unit ID</td></tr>
<tr><td>6</td></tr>
<tr><td>7</td></tr>
<tr><td>8</td></tr>
<tr><td>9</td><td>Spare</td></tr>
<tr><td>10</td><td>Spare</td></tr>
<tr><td>11</td><td>Authenticated Features Matrix Byte 0
<table>
<tr><td>Obs.</td><td>Obs.</td><td>Obs.</td><td>Obs.</td><td>Obs.</td><td>Obs.</td><td>Obs.</td><td>Obs.</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>12</td><td>Authenticated Features Matrix Byte 1
<table>
<tr><td>Obs.</td><td>Obs.</td><td>Obs.</td><td>Obs.</td><td>Obs.</td><td>PointerCept
0 - Inactive
1 - Active</td><td>Obs.</td><td>Obs.</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><td>13</td><td>Authenticated Features Matrix Byte 2<table>
<tr><td>Spare</td><td>Spare</td><td>Spare</td><td>Spare</td><td>Spare</td><td>Spare</td><td>Basic Driver Behavior
0 - Inactive
1 - Active</td><td>TDLT
0 - Inactive
1 - Active</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>14</td><td>Authenticated Features Matrix Byte 3</td></tr>
<tr><td>15</td><td>Authenticated Features Matrix Byte 4</td></tr>
<tr><td>16</td><td>Authenticated Features Matrix Byte 5</td></tr>
<tr><td>17</td><td>Authenticated Features Matrix Byte 6</td></tr>
<tr><td>18</td><td>Authenticated Features Matrix Byte 7</td></tr>
<tr><td>19</td><td>Authenticated Features Matrix Byte 8</td></tr>
<tr><td>20</td><td>Authenticated Features Matrix Byte 9</td></tr>
<tr><td>21</td><td>Authenticated Features Matrix Byte 10</td></tr>
<tr><td>22</td><td>Authenticated Features Matrix Byte 11</td></tr>
<tr><td>23</td><td>Authenticated Features Matrix Byte 12</td></tr>
<tr><td>24</td><td>Authenticated Features Matrix Byte 13</td></tr>
<tr><td>25</td><td>Authenticated Features Matrix Byte 14</td></tr>
<tr><td>26</td><td>Spare</td></tr>
<tr><td>27</td><td>Spare</td></tr>
<tr><td>28</td><td>Spare</td></tr>
<tr><td>29</td><td>Spare</td></tr>
<tr><td>30</td><td>Checksum</td></tr>
</table>

3.3.10 *Platform Manifest (old type)*

Generated as a reply to Firmware Platform Manifest request (refer to Command channel Command type 0x10, Status Type 0x10).

Byte number	Description	Value
1	‘M’	
2	‘2’	

Byte number	Description	Value
3	'C'	
4	Packet Type	0x0A
5	Unit ID	
6		
7		
8		
9	Processor family identifier	Please refer to: Cellocator Wireless Communication Protocol, Section 2.6.1: "The Firmware Platform Manifest" for detailed fields description.
10	Hardware interface and peripherals identifier	Please refer to: Cellocator Wireless Communication Protocol, Section 2.6.1: "The Firmware Platform Manifest" for detailed fields description.
11	Size of program memory (in 1024 bytes units; Before FW version 33o – 1024 words units) (LSB)	Please refer to: Cellocator Wireless Communication Protocol, Section 2.6.1: "The Firmware Platform Manifest" for detailed fields description.
12	Size of program memory (in 1024 bytes units; Before FW version 33o – 1024 words units) (MSB)	
13-14	Size of volatile memory (LSB)	Please refer to: Cellocator Wireless Communication Protocol, Section 2.6.1: "The Firmware Platform Manifest" for detailed fields description.
	Size of volatile memory (MSB)	
15-16	Size of internal non-volatile memory (LSB)	Please refer to: Cellocator Wireless Communication Protocol, Section 2.6.1: "The Firmware Platform Manifest" for detailed fields description.
	Size of internal non-volatile memory (MSB)	

Byte number	Description	Value
17-18	Size of external non-volatile memory (LSB)	Please refer to: Cellocator Wireless Communication Protocol, Section 2.6.1: "The Firmware Platform Manifest" for detailed fields description.
	Size of external non-volatile memory (MSB)	
19-20	External non-volatile memory type	Please refer to: Cellocator Wireless Communication Protocol, Section 2.6.1: "The Firmware Platform Manifest" for detailed fields description.
21	Hardware Version	Please refer to: Cellocator Wireless Communication Protocol, Section 2.6.1: "The Firmware Platform Manifest" for detailed fields description.
22-23	Reprogramming facility identifier (LSB)	Please refer to: Cellocator Wireless Communication Protocol, Section 2.6.1: "The Firmware Platform Manifest" for detailed fields description.
	Reprogramming facility identifier (MSB)	Please refer to: Cellocator Wireless Communication Protocol, Section 2.6.1: "The Firmware Platform Manifest" for detailed fields description.
24-25	Script language version (LSB)	Please refer to: Cellocator Wireless Communication Protocol, Section 2.6.1: "The Firmware Platform Manifest" for detailed fields description.
	Script language version (MSB)	Please refer to: Cellocator Wireless Communication Protocol, Section 2.6.1: "The Firmware Platform Manifest" for detailed fields description.
26-27	Current Firmware ID (LSB)	Please refer to: Cellocator Wireless Communication Protocol, Section 2.6.1: "The Firmware Platform Manifest" for detailed fields description.
	Current Firmware ID (MSB)	
28	Checksum	

3.3.11 *EEPROM Contents Dump Packet (Type 0x0B)*

The packet is sent as a response to a dump EEPROM contents command packet. It dumps the contents of the EEPROM in the specified region.

Data part: The data part of this packet has a variable size, but it is always at least five bytes long. The first 2 bytes contain the start address of the region whose contents is dumped. The second pair of bytes contains the number of bytes that are being dumped. According to it, the size of the whole data part may be determined. The rest of the bytes (at least one) are the contents of the specified region.

IMPORTANT: A zero value in the EEPROM region length (the third and fourth bytes of the data part) indicates a number of **1024 bytes!** Since a zero value is therefore nonexistent, all packets will contain at least one EEPROM byte, which means the data part will be at least five bytes long.

3.3.12 *Forwarding Modem's Response to AT command (Type 0x0C)*

After forwarding the AT Command (refer to Section 4.4.15), the unit will wait for modem response.

When a complete response is received or a Timeout Error occurred, the unit forwards the response to the serial interface (message type 0x0C) and reconnects communication if needed.

The structure of the outbound command from the unit is:

Byte	Data
1	'M'
2	'2'
3	'C'
4	0x0C
5	Unit ID
6	
7	
8	
9	Action error level
10	Number of response "lines" collected
11	Length of response field from modem (LSB)
12	Length of response field from modem (MSB)
	Command response, without CR/LF overhead, from modem for maximum 1000 bytes.

….	N bytes (where N = value of length field from bytes 12-13)
N+14	Checksum

3.3.12.1 Action Error Level Codes

Code	Response Error Levels — Command Was Sent
0	Command sent. All reply lines were collected.
1	Command sent. Timed out while waiting for response (any data received up to timeout is forwarded).
2	Command sent. Response is too long to store (partial response is forwarded).
	Immediate errors. Command was not sent
128	Non-zero lines specified, but timeout is zero.
129	Unit's forwarding buffers are currently occupied.
	Process errors. Aborted before command was sent.
160	Unit currently performs a non-breakable modem process.
161	Could not send the command (internal error).
162	Unable to perform preparatory modem disconnection, or modem is not responding.

3.3.13 *Forwarding of CAN Frames (Type 0xD)*

Forwarding of the CAN frames is performed by means of an outbound CAN frame forward message, type 13 (0Dh) when the CAN Debug mode is activated. Following is the message structure:

Byte	Data
1	'M'
2	'2'
3	'C'
4	0x0D
5	Unit ID
6	
7	
8	
9 - 13	CAN Frame Header
14 - 21	CAN Data

Byte	Data
22	Checksum

3.3.13.1 CAN Frame Header

Header – 5 bytes

Byte 1: Bits 3– 10 of standard ID.

Byte 2:

SID			r	x		EID	
0	1	2	3	4	5	6	7

SID – bits 0–2 of standard ID.

r – Remote request for standard frames.

x – Extended frame flag.

EID - bits 16–17 of extended ID.

Byte 3: Bits 8–15 of extended ID.

Byte 4: Bits 0–7 of extended ID.

Byte 5:

	R			DCL			
0	1	2	3	4	5	6	7

R – Remote request for extended frames.

DCL – length of data in bytes (0-8)

3.3.14 *Usage Counters Update (Type 0x15)*

This packet is triggered as a response to a Counter Write/Request Command (type 0x15) on an inbound channel,

3.3.14.1 Packet Structure

Byte	Data
1	'M'
2	'2'
3	'C'
4	Message type - 0x15
5	Unit ID
6	
7	

8	
9	Spare
10-13	Counter 1 data
14-17	Counter 2 data
18	Checksum

3.3.14.2 Counters Data Field Definition

Counter 2 value (minutes, 0-0xFFFFFF)			Counter 2 Input's number	Counter 1 value (minutes, 0-0xFFFFFF)			Counter 1 Input's number
Byte 18	Byte 17	Byte 16	Byte 15	Byte 14	Byte 13	Byte 12	Byte 11

3.3.14.3 Input Numbers Definition

Input Name	**Input Number (dec)**
Hardware: Olympic	
Shock	1
Ignition	5
Panic	6
Ext. Alarm Triggered	12
Ext. Alarm Armed	13
Hardware: Compact CAN	
Door	0
Shock	1
Ignition	5
Panic	6
Hardware: Compact Security/6 inp.fleet	
Door	0
Unlock2 Input	1
Ignition	5
Panic	6
Unlock Input	7
Lock Input	10

Input Name	Input Number (dec)
Hardware: Compact Fleet/LC	
Door	0
Shock	1
Ignition	5
Panic	6
COM RTS (pin 11)	7
Hardware: 370-50	
Door	0
Shock	1
Hood	2
Volume	3
Ignition	5
Panic	6
GP1	7
Arm	8
Disarm	9
Odometer	11
Unlock	12
Lock	13
Unlock2	14
Hardware: CelloTrack/ CelloTrack Power/ CelloTrack Output	
Tamper Switch	0
Push Button	1
GP input 1	2
GP input 2 (not for CelloTrack output)	3
Movement Sensor (Ignition)	5

3.3.15 *Auth Table Content (Packet Type 0x16)*

This packet is sent as a response to the Auth Table write/read command (Type 0x16); see the description in Section 4.4.18.

Byte	Data		
1	'M'		
2	'2'		
3	'C'		
4	Message type - 0x16		
5	Unit ID		
6			
7			
8			
9	Spare		Data part
10	Auth Table Index 0	Auth Table Index 1	
17	Auth Table Index 14	Auth Table Index 15	
19	Checksum		

3.3.16 *Shipment MODE status (Packet Type 0x18)*

This packet is sent as a response to Shipment mode Status request (Type 0x18), see description below.

The unit is capable to disconnect it's the backup battery in order to prevent an unnecessary discharge.

Once the shipment mode is activated the battery will be disconnected until the next connection of the main power supply.

NOTE: The shipment mode is activated by default. It means, that after complete power loss (both main and backup power supplies are disconnected) the battery will be disconnected (and reconnect only upon reconnection of the main power).

Byte	Data
1	'M'
2	'2'
3	'C'
4	Message type - 0x18
5	Unit ID
6	
7	
8	

Byte	Data
9	Shipment mode status Shipment mode active Shipment mode inactive
10	Checksum

3.3.17 *Debug Log (Type 0x19)*

This packet is sent as a response to Read Debug log command (Type 0x19), see description below.

Byte	Data
1	'M'
2	'2'
3	'C'
4	Message type - 0x19
5-8	Unit's ID
9-10	Payload length (in bytes)
11-N	Debug log
N+1	Checksum

3.3.17.1 Debug log

The debug log is organized as time-stamped events, 10bytes each.

Event's Info		Event Reason		10 msecs	Secs	Mins	Hours	Day	Month	Year
Byte 10	Byte 9	Byte 8	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0

3.3.18 *Firmware image (Type 0x1C)*

This packet is sent as a response to Firmware image request (Type 0x1C), see description below.

Byte	Data
1	'M'
2	'2'

Byte	Data
3	'C'
4	Message type - 0x1C
5-45	File name string, HEX, 40 bytes
46	Checksum

3.3.19 *Modular Platform Manifest (Type 0x1D)*

Only supported from fw30

Generated as a reply to a Modular Platform Manifest request (see Command channel, command 1D)

Byte	Description
1	'M'
2	'2'
3	'C'
5	Message type - 0x1d
6	Unit ID
7	Unit ID
8	Unit ID
9	Unit ID
10	Payload length
	Field 1 - Identifier
	Field 1 – Length of payload
	Field 1 - Payload
	…
	Field X - Identifier
	Field X – Length of payload (bytes)

	Field X - Payload
	Checksum

3.3.19.1 Fields Definition

The Modular platform manifest detailed fields description can be found in the "Cellocator Wireless Communication Protocol document. Please refer to Cellocator Wireless Communication Protocol, Outbound, Type 9 message: Modular Platform Manifest (Sub-Data 0x12).

3.3.20 *IICID Response (Type 0x28)*

CelloTrack3G: This packet is sent as a response to IICID query command.

Byte	Data
1	'M'
2	'2'
3	'C'
4	Message type - 0x28
5	Unit's ID
6	Unit's ID
7	Unit's ID
8	Unit's ID
9	Payload Length
10-29	IICID up to 20 ACSII bytes
30	Checksum

3.3.21 *PointerCept Base Station – Beacon Detected (Type 0x40)*

This message is sent from PointerCept base station upon one of the following scenarios:

- A beacon is detected
- Upon Status Request/Power Control from the server or mobile app (with beacon detect request)

Byte	Data
1	'M'
2	'2'
3	'C'
4	0x40
5	Unit ID (of the base station MCU)

6	
7	
8	
9	Numerator
10	End Unit ID (The MSB byte is reconstructed by the server based on the 3 bytes received from the beacon and the same ID. MSB byte is be sent as 0)
11	
12	
13	
14	Receiver ID (0-5) 7 - for a case in which more than receiver received the batch (e.g. in chase mode)
15	Average RSSI per batch (0-255)
16	Max RSSI per batch (0-255)
17	Min RSSI per batch (0-255)
18	RSSI standard deviation of the batch (0-255)
19	MER – Message Error Ratio between received number of messages in batch (Hop numerator sum) / and expected number of messages in batch (PL based) (%)
20	Indication of received hops 0-7 (bit indication per hop)
21	Indication of received hops 8-15 (bit indication per hop)
22	Indication of received hops 16-23 (bit indication per hop)
23	Indication of received hops 24-31 (bit indication per hop)
24	Indication of received hops 32-39 (bit indication per hop)
25	Indication of received hops 40-47 (bit indication per hop)
26	Indication of received hops 48-55 (bit indication per hop)
27	Indication of received hops 56-63 (bit indication per hop)
28	Indication of received hops 64-71 (bit indication per hop)
29	Indication of received hops 72-76 (bit indication per hop)
30	BS main CPU Vin: resolution will depend on the voltage divider - (Resolution/coefficient is 0.117647V per bit)
31	BS main CPU Batt. Voltage: resolution will depend on the voltage divider- (Resolution/coefficient is 0.01647V per bit)
32	Cello Status

	Unused	Unused	Unused	Unused	Unused	Unused	GPS Status 0 – Not Fixed 1 - Fixed	OTA Channel (GPRS Socket) 0 – Not Available 1 - Available
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
33	Transmission Reason (same as OTA Type 0 TR (byte 19) – refer to wireless protocol section 2.2.3.12 for details)							
34	Beacon packet type 00 – Location message 01 – Complementary information message 10 – PSP information message 11 - Reserved		Time till TX on frame end (seconds)					
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
35	Jamming 0 – Not Jammed 1 - Jammed	Resolution 0 – Seconds 1 - Minutes	Time till next TX on frame start (seconds/minutes)					
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
36	Modem PA Temperature							
37	Batch Numerator							
38								
39								
40								
41								
42								
43	Payload (depends on beacon message type)							
44								
45								
46								
47								
48								

49	Hop Numerator Detected by Base Station (used for chase mode only)
50	Receiver Assigned Group
51	Reserved
52	Reserved
53	Checksum

Payloads description:

Location message:

Byte	Data
38	Last GPS Fix
39	
40	Longitude
41	
42	
43	
44	Latitude
45	
46	
47	
48	Ground Speed (0-255 km/h)

Complementary information message:

Byte	Data
38	Unit's I/O status 1st byte
39	Unit's I/O status 2nd byte
40	Unit's I/O status 3rd byte
41	Unit's I/O status 4th byte
42	Mode 1 (from GPS)
43	Mode 2 (from GPS)
44	Number of satellites used (from GPS)
45	Speed direction: direction (angle) of the speed vector mapped into one byte. The span is between 0 to 2pi where each bit is represented by: 2pi *1000 / 255 [radians/1000].
46	Power level: 0 to 40 Volts in 0.15 volts per bit. This byte will Reflect the unit's power level. When the voltage is low it will reflect the batt. level, when external power is connected it shall reflect the external power voltage.
47	Modem PA Temperature
48	Reserved

PSP information message:

Byte	Data
38	PSP Dallas Byte 0
39	PSP Dallas Byte 1
40	PSP Dallas Byte 2
41	PSP Dallas Byte 3
42	PSP Dallas Byte 4
43	PSP Dallas Byte 5
44	Reserved
45	Reserved
46	Reserved
47	Reserved
48	Reserved

Notes:

- The Base Station always saves in buffer the last received Beacon detect message for each Receiver 0-5.
- If Beacon Detect Request message was sent, the base station MCU replies with last Beacon Status message for each receiver
- Bytes 33-37 contain the last Hop received information

3.3.22 *PointerCept Base Station – Status (Type 0x41)*

This message is sent from PointerCept base station upon Status Request/Power Control from the server (with status request).

Byte	Data
1	'M'
2	'2'
3	'C'
4	0x41
5	Unit ID (of the base station MCU)
6	
7	
8	
9	Numerator
10	Cello Status

	Unused	Unused	Unused	Unused	Unused	Unused	GPS Status 0 – Not Fixed 1 - Fixed	OTA Channel (GPRS Socket) 0 – Not Available 1 - Available
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

11 — Base station type/operation mode/power mode

Unused	Unused	Base station power mode 0 - Performance 1 - Power Conservation 2 – STBY		Unused	Base station operation mode 0 - Search 1 - Chase	Unused	Base station type 0 - Fixed 1 - Mobile
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

12 — Power State 1 (0 – Disabled, 1 – Enabled)

Unused	Unused	RCV 5	RCV 4	RCV 3	RCV 2	RCV 1	RCV 0
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

13 — Power State 2 (0 – Disconnected, 1 – Connected)

Unused	Unused	Unused	Unused	Unused	Unused	BT	USB
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

14 — Power State 3 (0 – Not active, 1 – Active)

Unused	Unused	Unused	Battery active (in case main power is under 9V)	Main power (above 9V)	Output	Charging	Ignition
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

15 — Bluetooth Connection State
0 – Reserved
1 – Disconnected
2 – Connected

<table>
<tr><td>16</td><td>HW Alarms 1 (0 – No Error, 1 – Error)
<table><tr><td>Unused</td><td>Cello communication</td><td>RCV 5</td><td>RCV 4</td><td>RCV 3</td><td>RCV 2</td><td>RCV 1</td><td>RCV 0</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>17</td><td>HW Alarms 2 (0 – No Error, 1 – Error)
<table><tr><td>Charger</td><td>Battery Low</td><td>Battery</td><td>Memory</td><td>General BIT</td><td>Cello via Battery Voltage Switch</td><td>BT</td><td>Cello via External Voltage Switch</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>18</td><td>Vin Voltage: Resolution is 0.117647V per bit</td></tr>
<tr><td>19</td><td>DC2DC (VDP_Mon) Voltage: Resolution is 0.1877345 (TBD)</td></tr>
<tr><td>20</td><td>Battery Voltage: Resolution is 0.01647V</td></tr>
<tr><td>21</td><td>NTC Temperature: Signed, resolution is 1°C per bit</td></tr>
<tr><td>22</td><td>RCV. RSSI 0</td></tr>
<tr><td>23</td><td>RCV. RSSI 1</td></tr>
<tr><td>24</td><td>RCV. RSSI 2</td></tr>
<tr><td>25</td><td>RCV. RSSI 3</td></tr>
<tr><td>26</td><td>RCV. RSSI 4</td></tr>
<tr><td>27</td><td>RCV. RSSI 5</td></tr>
<tr><td>28</td><td>Spare</td></tr>
<tr><td>29</td><td>Checksum</td></tr>
</table>

3.3.23 *PointerCept Base Station – Power Status (Type 0x42)*

This message is sent from PointerCept base station upon one of the following scenarios:

- Power On/Off
- Battery level gets below the value configured in "Battery Power Conservation Mode Low Threshold" parameter (address 3473)

Byte	Data
1	'M'
2	'2'
3	'C'
4	0x42
5	Unit ID (of the base station MCU)
6	
7	

8	
9	Numerator
10	Status Code 0 – Spare 1 – Power On – is sent immediately after turn on. All numerators are reset to 0. 2 – Power Off – is sent 5 seconds before going to Power Off mode, upon: • User manual turn off • OTA command (currently not supported) 3 - Battery Low – is sent when: • In Power Conservation mode • Ignition is off • Battery level gets below the value configured in "Battery Power Conservation Mode Low Threshold" parameter (address 3473) 4 - AHR – is sent 5 seconds before performing AHR
11	Spare
12	Checksum

3.3.24 *PointerCept Base Station – Location Request (Type 0x43)*

This message is initiated by the chaser mobile app when trying to locate a certain PointerCept end unit with server help (e.g. in cases of GPS signal jammed, unavailable, etc.). It is then forwarded by the PointerCept base station to the server.

Byte	Data
1	'M'
2	'2'
3	'C'
4	0x43
5	Unit ID (of the base station MCU)
6	
7	
8	
9	Numerator
10	End Unit Under Chase ID
11	
12	
13	
14	Command Code 0 – Spare

	1 – Location Request – reply with Chase Start\Location Update
15	Spare
16	Checksum

3.3.25 *PointerCept Base Station – User Registration/Connection Approval (Type 0x44)*

After performing BT pairing with base station, the mobile app user have to type a password in order to pass the login screen.

The PointerCept base station receives this password and compares it to its stored password (configurable by "Mobile App Password" parameter, address 1911-1919).

If the password is not correct, the PointerCept base station informs the mobile app (on BT channel) of failure with Connection Approval message:

Byte	Data
1	'M'
2	'2'
3	'C'
4	0x44
5	Unit ID (of the base station MCU)
6	
7	
8	
9	Numerator
10	Connection Approval Code 0 – Not Approved 1 – Approved
11	Spare
12	Checksum

If the password is correct, the PointerCept base station informs the mobile app (on BT channel) of success with Connection Approval message, and informs the server of user log-in to mobile app with User Registration message:

Byte	Data
1	'M'
2	'2'
3	'C'
4	0x44

5	Unit ID (of the base station MCU)
6	
7	
8	
9	Numerator
10	Spare
11	Spare
12	IMSI Byte 1
13	IMSI Byte 2
14	IMSI Byte 3
15	IMSI Byte 4
16	IMSI Byte 5
17	IMSI Byte 6
18	IMSI Byte 7
19	IMSI Byte 8
20	IMSI Byte 9
21	IMSI Byte 10
22	IMSI Byte 11
23	IMSI Byte 12
24	IMSI Byte 13
25	IMSI Byte 14
26	IMSI Byte 15
27	Spare
28	Spare
29	Spare
30	Checksum

3.3.26 *Data fields over serial*

This packet is used only by Cello-CANiQ, from release 38n and up, over Cable-replacement (using BT-Extender), outbound only and transmitted periodically according to user configuration (PL).

Byte	Data
1	'M'
2	'2'
3	'C'

4	0x3C
5	Unit ID
6	
7	
8	
9	Spare
10	Spare
11	Total length
12	Data field # (1)
13	Field length (1)
14	Data field (1)
…	
…	Data field # (2)
…	Field length (2)
…	Data field (2)
…	
…	…..
…	Spare
…	Checksum

3.3.27 *Modem FOTA Response (Type 0x50)*

This packet enables sending Modem FOTA process results, upon finish of Modem FOTA process, initiated by receiving Modem FOTA Command (packet 0x50).

Byte	Data
1	'M'
2	'2'
3	'C'
4	0x50
5	Unit ID
6	
7	
8	

9	Spare
10	Spare
11	Spare
12	Modem FOTA Process Status 0 – ACK (Process Ended Successfully) 1 – Modem Nack, General Error 2 – Modem Nack, Firmware Corrupted, CRC Error 3 – Modem Nack, Firmware Package Mismatch 4 – Modem Nack, Firmware Signature Failed 5 – Modem Nack, Authentication Failed 6 – Modem Nack, Out of Memory Resource 20 – FW (MCU) Nack, FTP Session Failed 21 – FW (MCU) Nack, Illegal FTP Directory or Non-Exist Directory 22 – FW (MCU) Nack, File Downloading Failed 23 – FW (MCU) Nack, Upgrade Failed 24 – FW (MCU) Nack, Upgrade Timeout Expired 30 – FW (MCU) Nack, General Error 31 – FW (MCU) Nack, FOTA Process Terminated due to Unit Reset
13	Spare
14	Spare
15	Modem Sub Version After Upgrade Attempt (1 byte, 0-255)
…	Modem Version After Upgrade Attempt (ASCII String, 1st byte is length)
…	Checksum

4 Inbound Packets (Received by Unit)

4.1 Overview

As explained in Section 2, inbound packets require handshaking prior to actual transmission. All inbound packets follow the same basic structure, as described here:

1. M2C synchronization string (for **3-wire interface only**)
2. Packet type code (see types below, 1 byte. Note that for a **4-wire interface (obsolete)**, it will be the first byte of the command).
3. Packet data (number of bytes dependent of packet type)
4. Checksum (1 byte).

All packets contain parts 2 and 4. The M2C header is sent only when the unit is working with a 3-wire interface. Another part that varies among the different packet types is part 3 (packet data) whose length is a function of the packet type, and possibly even the packet data itself. Some packet types do not contain a data part at all. The checksum field is the additive unsigned sum, modulo 8-bit, of all bytes in the packet, excluding the synchronization string (if any) and the checksum part itself. It can be expressed mathematically as:

$$Checksum = \mathrm{mod}\, 2^8 \sum_{i=1}^{1+N} Byte(i),$$

where *N* is the length of data part (which is zero, in the absence of a data part).

4.2 Handshaking Procedure (4-Wire Interface Only)

To send a packet to the unit, the following handshaking procedure must be followed:

- Raise the handshake (DCE-RTS) line (this line should be kept low by default).
- Wait until the unit replies with two bytes, ASCII "OK".
- Send the packet to the unit (without M2C synchronization string).

In a **3-wire interface** (as explained in Section 2) an "M2C" synchronization string is sent as a header of any command to the unit. Note: The "M2C" is not included in checksum calculation.

In a 4-wire interface (obsolete interface), before the unit can be sent with another packet, the handshake line must be lowered. The line can be lowered back after the "OK" is received from the unit (it is recommended to lower the line at this point).

After "OK" is sent by the unit, it waits for a packet for as long as 600 msec.

If no byte is received within this timeframe, the unit stops listening to the line, and the terminal is required to go through the whole handshake process again, in order to send a packet.

During the actual transmission of a packet to the unit, bytes should not be more than 2.3 msec apart (the equivalent of about 2-byte time at 9600 BPS). If the amount of time from the stop bit of a byte to the start bit of the next byte exceeds 2.3 msec, the unit stops listening to the line and discards all received data.

4.3 Packet Types Summary (Inbound)

The following packet types are hereby defined:

Type Code (Hex)	Type Code (Dec)	Packet Type
00	00	Status/information request packet.
01	01	Unit state change packet.
02	02	Reserved for manufacturer use.
03	03	Reserved for manufacturer use.
04	04	Inputs status request packet.
05	05	Read 16 bytes of manufacturing traceability data.
06	06	Write 16 bytes of manufacturing traceability data.
07	07	Reserved for manufacturer use.
08	08	Forward data to wireless channel command (Obsolete, replaced by type 23).
09	09	Authenticated Features Command
0A	10	GPS Data request.
0B	11	Dump EEPROM contents command packet.
0C	12	Write EEPROM command packet.
0D	13	Command to start learning Arm/Disarm signal correlation data (for security oriented firmware only).
0E	14	Logic override command.
0F	15	Outputs set command.
10	16	Status request command.
13	19	Forwarding AT Command from serial interface to modem.
14	20	CAN debug mode control.
15	21	Usage counters access command.
16	22	Authentication table read/write command.

Type Code (Hex)	Type Code (Dec)	Packet Type
17	23	Shipment mode activation command.
18	24	Shipment mode status request.
19	25	Read debug log command.
1A	26	Reserved for manufacturer use (debug purposes).
1B	27	
1C	28	Read firmware image info.
1D	29	Modular Platform Manifest request.
1F-22	31-34	Reserved for manufacturer use (debug purposes).
23	35	Forward data to wireless channel command.
…	…	
31	49	Reserved: OP Session Description query.
32	50	Cello-IQ and Cello-CANiQ DFD.
33	51	Reserved (CFE).
35	53	Voice Call Management.
37	55	Nano Status Interrogation
38	56	Reserved: Nano PT
39	57	Nano and CelloTrack-4: Request to send a distress event to server
3A	58	Reserved: Nano PT
40	64	PointerCept Base Station – Status Request/Power Control
41	65	PointerCept Base Station – Chase Start/End/Location Update
46	70	Pass through
47	71	Unit inactivation
48	72	Reserved for CelloTrack-Phoenix (LV300P)
49	73	Reserved for CelloTrack-Phoenix (LV300P)
50	80	Modem FOTA Command

The actual structure of each of these packet types is defined later in this document.

4.4 Detailed Inbound Packets Structures

4.4.1 *Status/Information Request Packet (Type 0x00)*

This packet requests a unit status/information packet (type 02h).

Data part: The packet does not contain a data part.

4.4.2 *Packet Change (Type 0x01)*

Unit state change

Byte	Data
1	'M'
2	'2'
3	'C'
4	Message type - 0x01
5	Unit's ID
6	Unit's ID
7	Unit's ID
8	Unit's ID
9	Data Byte 1: See Below
10	Data Byte 2: See Below
11	Checksum

Data Bytes description:

Data Byte 1 9th byte	Data Byte 2 10th byte	Command
0x00	0x00	Go to standby
0x01	0x01	Go to emergency mode
0x02	0x02	Reset • The following fields will be reset: The "Garmin Enabled", "Garmin Connected" and GSM hibernation indication bit flags, Message numerator, Unit's status, Current GSM operator report, Unit's mode of operation, I/O, Analog inputs, Driver ID /PSP Specific Data/Accelerometer Status, Last GPS Fix, Number of satellites, Longitude, Latitude, Altitude, Speed, Course, System time, System date.

		• The modem will be re-initialized, the GPRS connection restored. • The RAM buffer used for data forwarding will be reset • Configuration parameters will be reloaded from Configuration memory
0x03	0x03	Enter garage mode (security unit only)
0x04	0x04	Arm alarm (security unit only)
0x05	0x05	Release from emergency mode (does not stop the Siren, only stops emergency transmissions)

4.4.3 *Unit Inputs Status Request Packet (Type 0x04)*

This packet triggers a unit inputs status packet.

Data part: The packet does not contain a data part.

4.4.4 *Read 16 bytes of Manufacturing Traceability Data (Type 0x05)*

This package is designed to allow reading of production tractability data, stored by the production tester. It triggers a type 5 response of the unit, including 16 bytes, stored during manufacturing process.

Data part: The packet does not contain a data part.

Header			**Type**	**Unit ID (4 bytes)**				**CS**
M	2	C	05					

4.4.5 *Write 16 bytes of Manufacturing Traceability Data (Type 0x06)*

This package is designed to allow writing of production tractability data during manufacturing. It triggers an ACK packet in the event of success.

Header			**Unit ID**				**Type**	**Data (16 bytes)**															**CS**
M	2	C					06																	

Data part: the packet contains 16 bytes of manufacturing a data part.

4.4.6 *Forward Data to Wireless Channel Command (Type 0x08, obsolete)*

This packet is still supported for backward compatibility reasons, but not recommended for future development. Use message type 0x23 for data forwarding instead.

This packet causes the unit to repeat the attached data to the wireless channel, using a special wireless message (refer to wireless channel documentation for further information).

Data part: The data part of this packet has a variable size, but it is always at least two bytes long. The first byte should contain the length of the data to forward, in bytes. The rest of the data part bytes should contain the actual data that is to be forwarded to the

wireless channel, which should be sent immediately after the length byte. The number of bytes should be as specified in the first byte.

NOTE: The data size byte (the first byte of the data part) may have any value from one (including one) to 82 decimal (including 82). Any value outside this span will cause the unit to discard the packet. The number of bytes that can be forwarded with a single command is therefore limited to 82.

4.4.7 *Authenticated Features Command (Type 0x09)*

This message enables query/activation/de-activation of features in the unit. It contains the desired features codes.

On query command, there will be no feature codes.

On activation/de-activation command the unit will reply with ACK (message type 0x00) or NACK (message type 0x01), while on query command the unit will reply with Authenticated Features Query Response (message 0x09).

<table>
<tr><th>Byte</th><th>Data</th></tr>
<tr><td>1</td><td>'M'</td></tr>
<tr><td>2</td><td>'2'</td></tr>
<tr><td>3</td><td>'C'</td></tr>
<tr><td>4</td><td>0x09</td></tr>
<tr><td>5</td><td rowspan="3">Unit ID</td></tr>
<tr><td>6</td></tr>
<tr><td>7</td></tr>
<tr><td>8</td><td></td></tr>
<tr><td>9</td><td>Control Byte
<table>
<tr><td>Spare</td><td>Spare</td><td>Spare</td><td>Spare</td><td>Spare</td><td>Spare</td><td colspan="2">Command Code
0 - Query
1 - Activation
2 - De-Activation</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>10</td><td>Spare</td></tr>
<tr><td>11</td><td>Number of Feature Codes
(0 for query command, 1-8 for activation/de-activation)</td></tr>
<tr><td>12</td><td>C[0]</td><td rowspan="3">Feature Code 1</td></tr>
<tr><td>13</td><td>C[1]</td></tr>
<tr><td>14</td><td>C[2]</td></tr>
</table>

15	C[3]	
16	Spare	
17	Spare	
18	Spare	
19	Spare	
20	Spare	
21	Spare	
22	C[0]	Feature Code 2
23	C[1]	
24	C[2]	
25	C[3]	
26	Spare	
27	Spare	
28	Spare	
29	Spare	
30	Spare	
31	Spare	
…	…	
	C[0]	Feature Code n
	C[1]	
	C[2]	
	C[3]	
	Spare	
	Spare	
	Spare	
	Spare	
	Spare	
	Spare	
…	Checksum	

4.4.8 *GPS Data Request Packet (Type 0x0A)*

This packet triggers a GPS Data packet.

Data part: The packet does not contain a data part.

4.4.9 *Dump EEPROM Contents Command (Type 0x0B)*

This command makes the unit dump the contents of a specified region of the internal EEPROM.

Data part: The data part of this packet is 8 bytes long. The starting four bytes should contain the serial number of the unit (if the value does not match the serial number, the packet is discarded and not handled). The fifth and sixth bytes should contain the hex value of start address of the region whose contents is to be dumped, while the seventh and eighth bytes should contain the length of it (hexadecimal value).

IMPORTANT: The full content of a unit's EEPROM is 1024 bytes, starting from address 0x00.

4.4.10 *Write EEPROM Command Packet (Type 0x0C)*

This packet forces the unit to write the attached data to a certain region of the internal EEPROM. The region is specified by start address and length.

Data part: the data part of this packet has a variable size, but it is always at least nine bytes long. The starting four bytes should contain the serial number of the unit (if the value does not match the serial number the packet is discarded and not handled). The fifth and sixth bytes should contain the start address of the region whose contents is to be written, the seventh and eighth bytes should contain the length of it.

In addition to the constant eight bytes, the data part should contain the actual data that is to be written to the EEPROM, which should be sent immediately after the constant six bytes. The number of bytes should be as specified in the seventh and eighth bytes.

NOTE: The region size byte (the sixth byte of the data part) may have any value from one (including one) to 48 decimal (including 48). Any value outside this span will cause the unit to discard the packet. The quantity of bytes that can be programmed with a single command is therefore limited to 48.

CAUTION: Portions of the EEPROM are reserved for use by internal mechanisms (namely the hopping code remote control reception mechanism), and under no circumstances may be written to, as doing so may damage the functionality of these mechanisms. Please refer to the Programming Manual document for further information.

4.4.11 *Command to Start Learning Arm/Disarm Signal Correlation Data (Type 0x0D)*

This packet triggers a process of learning arm/disarm procedure from corresponding inputs.

Data part: The packet does not contain a data part.

0x00 – Learn Lock Signal

0x01 – Learn Unlock Signal

0x02 – Learn Unlock 2 Signal

0xFF – Clear Learned Signals from EEPROM

4.4.12 *Logic Override Command (Type 0x0E)*

Logic Override Command affects unit behavior.

4.4.12.1 Packet Structure

Byte	Data
1	'M'
2	'2'
3	'C'
4	Message type - 0x0E
5	Unit ID
6	
7	
8	
9	Spare
10	Override Value
11	Checksum

4.4.12.2 Data Specification

Override Value

- '0' for normal unit logic/behavior.
- '1' for logic halt mode (for test bench).

In this mode the unit will become completely passive and will cease any automated behavior. Outputs will not change automatically and triggering of sensors will not cause any reaction.

4.4.12.3 Result

ACK/NACK message will be sent according to success of command execution.

4.4.13 *Outputs Set Command (Type 0x0F)*

Outputs Set Command affects unit's outputs.

4.4.13.1 Packet Structure

Byte	Data
1	'M'
2	'2'
3	'C'
4	0x0F
5	Unit ID
6	
7	
8	
9	Spare
10	Action mask bitmap (which bits will be effected)
11	
12	
13	
14	Desired outputs state bitmap (What will be the state of the relevant bits, only the ones that are '1' in the "Action mask bitmap" field) For each bit: 0 – Inactive 1 - Active
15	
16	
17	
18	Checksum

4.4.13.2 Data Specification

Action mask bitmap defines the outputs that should change (bit value '1') and which should not change (bit value '0') as a result of this command. The Desired output state bitmap defines the desired logical level of the outputs. Only outputs whose respective action mask bit is '1' will be set with the appropriate value.

Bits allocation is for **both** bitmaps: the first 2 bytes contain the same data as the 3rd and 4th bytes of the I/O status field in the Cellocator wireless protocol. The last 2 bytes are spare.

Note: Not all outputs exist in all products.

Byte 10 (and 14):

CFE out 5	CFE out 4	CFE out 3	CFE out 2	GPS power (report only)	Grad. Stop (PWM immobilizer)	Siren	CFE out 1
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Byte 11 (and 15):

Charger (report only) 0 – Not charging 1- Charging	CFE out 6	Standard Immobilizer	Unused	Blinkers (in CelloTrack-T called "Output")	Unused		LED out
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

4.4.13.3 Result

ACK/NACK message will be sent according to success of command execution.

4.4.14 *Status Request Command (Type 0x10)*

Status Request command solicits a transmission of a message from unit.

4.4.14.1 Packet Structure

Byte	Data
1	'M'
2	'2'
3	'C'
4	0x10
5	Unit ID
6	
7	
8	
9	Spare
10	Status Type
11	Checksum

4.4.14.2 Data Specification

Status Type

Code	Description
0x08	Master Unit Status Request Command

0x0A	Platform Manifest Request Command

4.4.14.3 Result

The appropriate message will be transmitted by the unit.

4.4.15 *Forwarding AT Command from Serial Interface to Modem (Type 0x13)*

The unit now supports forwarding an AT Command from a serial interface to the modem (message type 13h). In order to forward an AT Command, the unit disconnects communication.

The structure of the inbound command to the unit is:

Byte	Data
1	'M'
2	'2'
3	'C'
4	Message type - 0x13.
5	Unit ID.
6	
7	
8	
9	Max response collection time, 50msec resolution (allows up to 12.75 sec).
10	Number of response "lines" to collect.
11	Spare.
12	Spare.
13	Length of command to modem.
….	AT command to modem (Including CR if required). N bytes (where N = value of byte from length field from point 7 above).
N+10	Checksum.

Following is the response collection logic

Timeout	Lines	Effect
Zero	Zero	Command is sent, response is not collected.
Zero	Nonzero	Illegal combination, error generated (128).

Nonzero	Zero	Up to 256 lines to collect, timeout as specified.
Nonzero	Nonzero	Max lines to collect as specified, timout as specified.

4.4.16 *CAN Debug mode control (Type 0x14)*

4.4.16.1 Overview

CAN debug mode provides a simple "sniffing" capability of the CAN bus, by forwarding received CAN frames from the CAN bus to the wire interface.

4.4.16.2 Caveats

This is not a fully-fledged sniffer, but rather a simple diagnostics tools. There are known weaknesses in this feature:

- Forwarding Throughput is limited to the 9600bps dictated by the serial port.
- Frame loss will occur on frames that are transmitted too closely to frames that are in the process of being forwarded. The mechanism properly handles frames that are more loosely inter-spaced.
- Occasionally, CAN frames forwarding may be momentarily disrupted due to GPS communication.

4.4.16.3 Suggested Uses

- Non-critical collection of data, to provide an insight into the messages available on a specific bus.
- Simple test to determine:
  - Whether the unit's CAN interface is functioning well.
  - Whether the unit is properly connected to the CAN bus, and set up correctly.

4.4.16.4 Command Structure

Byte	Data
1	'M'
2	'2'
3	'C'
4	Message type – 0x14
5	Unit ID
6	
7	
8	
9	Control Byte
10	Spare

Byte	Data
11	Spare
12	Checksum

4.4.16.5 Control Byte

Value	Effect
0	Disable CAN debug mode
1	Enable CAN debug mode

NOTE: While CAN debug mode is active, normal processing of CAN frames is disabled. This means CAN filters and triggers are not processed. Instead, the frames are immediately forwarded to wire interface.

4.4.17 *Usage Counter Write/Request Command (Type 0x15)*

The purpose of this feature is to count "High State" time of pair of inputs, for example to report total engine hours of some machine. The inputs, whose "High State" time is counted, are selectable by programming. Each of two timers can be assigned to a specific input, including the option to assign both timers into the same input.

Each input, including ignition, supports this "High State" time calculation. The value of the measured time from each input is stored in RAM (protected, not erased on software reset, 24 bits each parameter, not part of configuration memory), with a resolution of minutes.

The unit makes mathematical rounding on partial minutes: (1:29 will be counted as 1 minute and 1:30 and above as 2 minutes).

Once a day, the content of both usage counters is backed up on the especial address of non-volatile memory.

The timer processes time counting (from the value stored in RAM), each time that the logical level of the appropriate input changes from "Low to high". The Timer stops time counting when the logical level of the input changes from "High" to "Low".

The RAM values of Usage counters is automatically updated on each RS232 and OTA “Counter’s Set” commands.

4.4.17.1 Command structure

Byte	Data
1	'M'
2	'2'
3	'C'
4	Message type - 0x15
5	Unit ID
6	

Byte	Data
7	
8	
9	Control Byte
10	Spare
11-13	Counter 1 data
14-16	Counter 2 data
17	Checksum

4.4.17.2 Control Byte Definition

Structure

Unused	Action bits	
Bits 2-7	Bit 1	Bit 0

4.4.17.3 Action Bits Definition

Action bits	Bit 1	Bit 0
Read counters data	0	0
Write counter 1	0	1
Write counter 2	1	0
Write counter 1 and 2	1	1

4.4.17.4 Counters Data Field Definition

Counter 2 value (minutes, 0-0xFFFFFF)			Counter 1 value (minutes, 0-0xFFFFFF)		
Byte 16	Byte 15	Byte 14	Byte 13	Byte 12	Byte 11

NOTE: If the Control byte is zero (request command) the "Counters data" field is don't care and sent as zeros.

4.4.18 *Command Authentication (Type 0x16)*

The unit is capable of providing protection against unapproved command attempts. In every incoming OTA message to the unit (like command, acknowledge and so on) a unique code is provided which is verified by the unit. If the code is not verified as authentic, the unit will not perform / acknowledge the command.

The Command Authentication is programmable and should be enabled by the unit's programming to be activated.

If enabled, the unit checks a valid 4-byte authentication code in bytes 11–14 of every inbound OTA message. The inbound message with invalid authentication code is declined by the unit; the unit does not respond to such a command and does not perform it.

The 4 bytes authentication code is generated as a function of two variables:

- Unit's ID
- 8 bytes Auth Table, stored in the EEPROM of the unit and concurrently in the communication center application.

The OTA Auth. table modification will only be accepted by the unit if the Command Authentication feature is **DISABLED**.

The following are default values of the Authentication Table.

Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Value	2	15	7	9	12	1	4	6	8	3	11	14	0	5	10	13

4.4.18.1 Auth Table Write/Read Command Structure

This packet is sent to the unit in order to access The Auth Table values (read, write or modify).

Byte	**Data**		
1	'M'		
2	'2'		
3	'C'		
4	Message type - 0x16		
5	Unit ID		
6			
7			
8			
9	Action (see table below)		Data Part
10	Spare		
11	Auth Table Index 0	Auth Table Index 1	
18	Auth Table Index 14	Auth Table Index 15	
19	Checksum		

4.4.18.2 Action Byte

Value	Meaning	Remarks
0	Read Auth table from EEPROM	bytes 11-18 of the command are "don't care".
1	Write Auth table to EEPROM,	bytes 11-18 of the command will include the values to be programmed.
2-7	Reserved	

NOTE: Reset is required in order to apply OTA Authentication table modification.

4.4.19 *Shipment Mode Control Command (Type 0x17)*

This packet is sent when it is required to activate of reactivate Shipment mode (battery disconnect).

The unit is capable to disconnect it's the backup battery in order to prevent an unnecessary discharge.

The command will only be accepted while the unit is connected to the external power.

Once the shipment mode is activated the battery will be disconnected until the next connection of the main power supply.

The unit will wake itself up and start normal operation (including backup battery maintenance) upon reconnection to the external power.

NOTE: The shipment mode is activated by default. It means, that after complete power loss (both main and backup power supplies are disconnected) the battery will be disconnected (and reconnect only upon reconnection of the main power).

Command Structure

Byte	Data
1	'M'
2	'2'
3	'C'
4	Message type - 0x17
5	Action (see table below)
6	Checksum

4.4.19.1 Action Byte (Byte 1)

Value	Meaning
0	Switch on shipment mode
1	Switch off shipment mode

4.4.19.2 Response

Upon successful execution of the command, a generic ACK packet (outbound type 0) will be sent. In the event of a problem, a NACK (outbound type 1) will be sent.

4.4.20 *Shipment Mode Status Request (Type 0x18)*

See description of shipment mode in the previous command (type 0x17).

4.4.20.1 Command Structure

Byte	Data
1	'M'
2	'2'
3	'C'
4	Message type - 0x18
5	Checksum -0x18

4.4.20.2 Response

Upon successful execution of the command, an outbound type 18h (shipment mode status) will be sent.

4.4.21 *Read Debug Log Command (Type 0x19)*

This packet is sent when debug log is required.

Byte	Data
1	'M'
2	'2'
3	'C'
4	Msg type - 0x19
5	CheckSum (0x19)

4.4.21.1 Response

Upon successful execution of the command, an outbound type 19h (Debug log) will be sent.

4.4.22 *Firmware Image Info Request (Type 0x1C)*

This packet is sent when Firmware Image Info (textual description of the firmware) is required.

Byte	Data
1	'M'
2	'2'
3	'C'
4	Message type - 0x1C
5	Checksum -0x1C

4.4.22.1 Response

Upon successful execution of the command, an outbound type 0x1C (Firmware Image Info) will be sent.

4.4.23 *Modular Platform Manifest Request (Type 0x1D)*

Supported from fw30 only

This command causes the unit to generate a Modular Platform Manifest message (0x1D). The message will contain the data fields as per the specification in a command

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

Byte	Data
1	'M'
2	'2'
3	'C'
4	Message type - 0x1d
5	Bit Map 0
6	Bit Map 1
7	Bit Map 2
8	Bit Map 3
9	Bit Map 4

Byte	Data
10	Bit Map 5
11	Checksum -0x1c

The Bit Map sent as Modular Message request has the same structure as the corresponding OTA modular Platform Manifest Request. See: Cellocator Wireless Communication Protocol: Section 3.7.13: Modular Platform Manifest Request (Sub-Data Type 0x12)

Bit Map Byte 0 (Byte 5)

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

Bit Map Byte 1 (Byte 6)

Fw Upgrade Date/Time	Initial Power up Date/Time	GPS firmware	GPS Type	Modem firmware	Modem type	Current Hardware ID number	Current Firmware ID number
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit Map Byte 2 (Byte 7)

Head Version Base Request	IMSI/IMEI Request	VIN Request	DFD / SD card version Request	Bootloader 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

Bit Map Byte 3 (Byte 8)

Maintenance Configuration	Modem Firmware Sub Version	PIN#8 HW selected function	SIM ICCID	BT Extender Identification	Size of Internal Non-Volatile Memory	Version Information (Cello-(CAN)IQ)	Nano Sensors
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit Map Byte 4 (Byte 9)

Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Silicon Labs 4632 receiver version	BT version number
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit Map Byte 5 (Byte 10)

Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

4.4.24 *Long Data Forwarding Message (Type 0x23)*

This packet causes the unit to repeat the attached data to the wireless channel, using a special wireless message (refer to wireless channel documentation for further information).

Data part: The data part of this packet has a variable size, but it is always at least two bytes long. The first 2 bytes should contain the length of the data to forward, in bytes.

The rest of the data part bytes should contain the actual data that is to be forwarded to the wireless channel, which should be sent immediately after the length byte. The number of bytes should be as specified in the first byte.

Byte	Data
Byte 1	'M'
Byte 2	'2'
Byte 3	'C'
Byte 4	Message type – 0x23
Byte 5	Payload length Multi-byte field, little endian, Valid values 1-512. Values out of the valid range will result in a response of message type 02 with NACK flag) and dropping the payload.
Byte 6	
Bytes 7 to 7+X	The payload (X bytes)
Byte 8+X	Check Sum

4.4.24.1 Acknowledge to type message 0x23

The unit shall reply by Status/Information Packet (type 0x02)

4.4.25 *IICID query (Type 0x28)*

Cellotrack3G: This packet query for IICDID number (See outbound for response).

Byte	Data
Byte 1 (MSB)	'M'
Byte 2	'2'
Byte 3	'C'
Byte 4	Message type – 0x28
Byte 9	0x28

4.4.26 *Voice Call Management (Type 0x35)*

This packet causes the unit to repeat the attached data to the wireless channel, using a special wireless message (refer to wireless channel documentation for further information).

Data part: The data part of this packet has a variable size, but it is always at least two bytes long. The first 2 bytes should contain the length of the data to forward, in bytes.

The rest of the data part bytes should contain the actual data that is to be forwarded to the wireless channel, which should be sent immediately after the length byte. The number of bytes should be as specified in the first byte.

Byte	Data	
1	'M'	
2	'2'	
3	'C'	
4	Message type - 0x35	
5	Unit ID	
6		
7		
8		
9	Action Byte (see table below)	Data Part
10	Spare	
11	Action-specific data- 2 Options (10 bytes) Action-specific Data for "Originate Call" Action Action-Specific Data for "Control Volume" Action	
20		

Byte	Data
21	Checksum

4.4.26.1 Action Byte (Byte 1)

Value	Meaning	Action-Specific data
0	Disconnect all calls	Ignored
1	Originate a call to the number specified in this packet Note: the unit will reply only after reception of the response from the modem.	Target number
2	Reject incoming call	Ignored
3	Answer first call	Ignored
4	Infrastructure (currently not supported) Switch calls (toggle between waiting and active calls).	Ignored
5	Query status	Ignored
16	Control volume (The unit should process this command same way as it would process volume control by input)	Volume level

4.4.26.2 Action-specific Data for "Originate Call" Action

Byte number	Description
11	Digits in number (determines length of number in chars).
12	Type of address (TOA.)
13-20	Number in BCD lower nibble format (first digit in lower nibble).

4.4.26.3 Type of Address (Byte 12)

91h (145) – International numbering scheme (contains the character "+")

81h (129) – National numbering scheme (local number)

Example:

In order to dial +972-54-123456

Number of Digits = 0Bh

Type of Address (TOA) = 91h

The dialed number in BCD:

79h 52h 14h 32h 54h F6h FFh

4.4.26.4 Action-Specific Data for "Control Volume" Action

Byte number	Description
11	Zero level means mute request "1"- raise volume level "2"- decrease volume level
12-20	Unused.

4.4.26.5 Response

Upon successful execution of the command, a generic ACK packet (outbound type 0) will be sent. In the event of a problem, a NACK (outbound type 1) will be sent.

4.4.27 *Serial Upload Mode Control and Status (Type 0x2C)*

4.4.27.1 Serial upload mode by command from serial port

Cello shall support "Serial upload" mode.

<table>
<tr><th>Byte's number</th><th>Byte's number</th></tr>
<tr><td>1</td><td>'M'</td></tr>
<tr><td>2</td><td>'2'</td></tr>
<tr><td>3</td><td>'C'</td></tr>
<tr><td>4</td><td>0x2C</td></tr>
<tr><td>5</td><td rowspan="4">Unit ID</td></tr>
<tr><td>6</td></tr>
<tr><td>7</td></tr>
<tr><td>8</td></tr>
<tr><td>9</td><td>Control byte
<table>
<tr><td colspan="6">Reserved</td><td>0 – Request status
1 – Command (see bit 0)</td><td>0 – Stop
1 - Start
Only if bit 1 is 1</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>10</td><td>Spare</td></tr>
<tr><td>11</td><td>CS</td></tr>
</table>

The message will be acknowledged by outbound message 0x2C

4.4.27.2 Serial Upload Mode Status Response

Cello shall support "Serial upload" mode.

<table>
<tr><th>Byte's number</th><th>Byte's number</th></tr>
<tr><td>1</td><td>'M'</td></tr>
<tr><td>2</td><td>'2'</td></tr>
<tr><td>3</td><td>'C'</td></tr>
<tr><td>4</td><td>0x2C</td></tr>
<tr><td>5</td><td>Control byte
<table>
<tr><td>0</td><td>ACK to Request Status: The unit is NOT in Serial upload mode</td></tr>
<tr><td>1</td><td>ACK to Request Status: The unit is in Serial upload mode</td></tr>
<tr><td>2</td><td>ACK: All data is successfully uploaded</td></tr>
<tr><td>3</td><td>NACK: No data to upload (log is empty)</td></tr>
<tr><td>4</td><td>NACK: Comm. failure, timeout expired.</td></tr>
<tr><td>5</td><td>NACK: Process aborted by command</td></tr>
<tr><td>6</td><td>NACK: Process aborted by reset</td></tr>
<tr><td>7-255</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>6</td><td>Spare</td></tr>
<tr><td>7</td><td rowspan="3">Number of messages in the buffer</td></tr>
<tr><td>8</td></tr>
<tr><td>9</td></tr>
<tr><td>10</td><td>CS</td></tr>
</table>

The same message type is issued as a reply to Serial upload status request (0x2D).

4.4.27.3 Entrance to Serial Upload Mode

By command only.

If the unit is in hibernation upon entrance into this mode, it shall not wake up its modem or/and GPS. This feature doesn't work when Cello is in Full Hibernation with UART On.

4.4.27.4 Exit from Serial Upload Mode

By serial command.

After reset, complete emptying of the buffer, and upload failure.

4.4.27.5 Halt OTA Activity

If upon entrance to Serial upload the modem is on - the unit shall halt all OTA activity and restore it only upon leaving this mode (if not in hibernation).

If GSM peek occurs while the unit is Serial upload mode - this GSM peek will be postponed to the end of Serial upload mode.

4.4.28 *Serial Upload Mode DataFlow (Type 0x2D)*

While in Serial upload mode the unit is uploading its event's log through serial port.

The upload is very similar to a process, normally happening OTA:

The unit is sending up to 16 messages from its "ready to send" buffer and expecting to get acknowledge for each one of them. Acknowledged messages are erased from unit's log, unacknowledged are resent.

The acknowledge timeout is hardcoded 1 second.

The message is in the same format, as it is sent OTA encapsulated in the following frame:

4.4.28.1 Serial Upload - Data Packet

Byte's number	Byte's number
1	'M'
2	'2'
3	'C'
4	0x2D
5	Length of Payload, N
6	
7-(N+7)	Payload: OTA message (70 bytes for message type 0 or variable for data forwarded from Garmin)
N+8	CS

The serial ACKs are in same format as the ACK of OTA protocol, therefore they use the same interfaces – the ACK is sent per message numerator of OTA message.

4.4.28.2 Serial Upload - Data Packet Ack

Byte's number	Byte's number
1	'M'
2	'2'
3	'C'

Byte's number	Byte's number
4	0x2D
5	Length of Payload, N
6	
7-(N+6)	Payload: OTA message (28 bytes for ACK (message type 4) or variable for data forwarded from Garmin)
N+7	CS

4.4.28.3 Process Status message

Upon the end of the upload process, the unit will issue 6.25.2 Serial upload mode Status response

4.4.29 *Cello-IQ CSA Serial Support (Type 0x32)*

4.4.29.1 Overview

Cello-IQ uses the M2C serial protocol to manage multiple sessions between external serial entity and Cello-IQ internal applications. Internal Cello-IQ applications are identified by their handler Id (TPA Handler). Each application can support multiple sessions identified by a corresponding session ID.

All the Cello-IQ messages are encapsulated under M2C code number 50.

The current Cello-IQ supports the following sessions:

Session Number	Session type	Description
0	CSA application session	This session will carry application commands and responses to and from the Cello-IQ driver behavior application.
1	DFD session	This session carries DFD (Driver Feedback Display) messages to and from the DFD device. The DFD session is used for controlling the visual and audible driver's feedbacks. DFD control messages are sent every 200mSec.

4.4.30 *General Cello-IQ Header Format (Module 50)*

This module is intended to forward data to/from Cello-IQ application through the COM port. Note that the structure of the command is identical in both directions: to and from CSA. The application messages described in the following sections are sent via the payload block of this header.

1	'M" - 4D	Synchronization string
2	"2" - 32	
3	"C" - 43	
4	Module ID – 50 (dec)	
5	TPA Handler – Always zero	
6		
7	Session number	
8	N- Length of data forwarded to CSA (from byte 10 to N+13) (Two bytes)	
9		
10	Payload Data to be forwarded to CSA (N bytes), including the TPA Command IDs	
N+13	Check Sum – Legacy M2C terminating CheckSum	

The payload section will carry data modules with the following structure:

	Command ID
	Module Length
	Module data

4.4.30.1 Summary of CelloIQ Command Codes

Session	Command Code	Description
0	42	Calibration Matrix Set command through COM port
0	43	Response to Calibration Matrix Set command

Session	Command Code	Description
0	21	Raw Data Recording Control command
0	22	Response to Raw Data Recording Control command
0	45	Query Trip/Crash list (inbound)
0	45	Trip/Crash list Response (outbound)
0	47	Uploaded Chunk response ACK (inbound)
0	48	Uploaded File Name and header request (inbound)
0	48	Uploaded File Name and header Response (outbound)
1	50	DFD: Sound player control command
1	51	DFD: Response to player control command
1	52	DFD: LEDs control command
1	53	DFD: Response to LEDs control command
1	54	DFD: Maintenance Command
1	55	DFD: Status (also response to Maintenance Command)
1	56	DFD: Serial Number Read/Set Command
1	57	DFD: Response to Serial Number Read/Set Command
1	58	DFD: Default Audio String Read/Set Command
1	59	DFD: Response to Default Audio String Read/Set Command
1		DFD: Sound player control command

4.4.30.2 Calibration Matrix Set Command Through COM Port

This command utilizes infrastructure of module 50 to deliver Calibration Matrix Set command to CSA. Module Length will be 2 in case of Read Calibration Matrix, 23 in case of write Calibration Matrix (as per Action Byte).

The calibration data consists of 30 signed 2-byte parameters. For example the matrix below:

9:9:-250		
16384:0:0	0:16372:592	0:-592:16372
16372:0:592	0:16384:0	-592:0:16372
16165:2671:0	-2671:16165:0	0:0:16384

NOTE: The diagonal of the matrix is always the same (16384:0:0, 0:16384:0, 0:0:16384), so we to keep only 21 prams as shown below.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Index	1	2	3						
Data	9	9	-250						
Index				4	5	6	7	8	9
Data	16384	0	0	0	16372	592	0	-592	16372
Index	10	11	12				13	14	15
Data	16372	0	592	0	16384	0	-592	0	16372
Index	16	17	18	19	20	21			

	Command ID – 42 Calibration Matrix Set /Read command
	Module Length -3 (2 in case of read, 23 in case of write)
	Action byte – 0 read, 1 write
	Spare
	Index 1 (16 bits)
	Index 2 (16 bits)
	Index 21(16 bits)

4.4.30.3 Response to Calibration Matrix Set Command

This module is issued in serial message 50 (dec) as a response to Calibration Matrix Set command described above. Module length is 22:

	Command ID – 43 Response to Calibration Matrix Set /Read command
	Module Length -22
	Spare (zeros)
	Index 1 (16 bits)
	Index 2 (16 bits)
	Index 21 (16 bits)

4.4.30.4 Raw Data Recording Control command

Byte number	Description	Content
N	Module ID-21	
N+1	Module length	5
N+2	Control byte	0 – Start (with duration listed below) 1 - Stop
N+3	Raw Logger Mode duration	In resolution of 2 second/bit. 0 – Unlimited, 1-FFFF maximum value 36 hours.
N+4		
N+5	Provision: GPS recording resolution	Shall be sent as: 1 Resolution of seconds
N+6	Provision: Accelerometer recording resolution	Shall be sent as: 5 Resolution of 10msec

4.4.30.5 Response to Raw Data Recording Control Command

Byte number	Description	Content
N	Module ID-22	
N+1	Module length	5
N+2	Operational Mode	Same as in module 30
N+3	Raw Logger Mode duration (only relevant for Raw Data mode)	In resolution of 2 second/bit. 0 – Unlimited, 1-FFFF maximum value 36 hours.
N+4		
N+5	Provision: GPS recording resolution	Shall be sent as: 1 Resolution of seconds
N+6	Provision: Accelerometer recording resolution	Shall be sent as: 5 Resolution of 10msec

4.4.30.6 File Upload Through COM Port

In some cases, like upon damage due to crash, OTA data upload is impossible. In such a case it shall be possible to upload files from COM port of Cello unit using Cellocator Serial Protocol. The following procedures should be performed to upload a file:

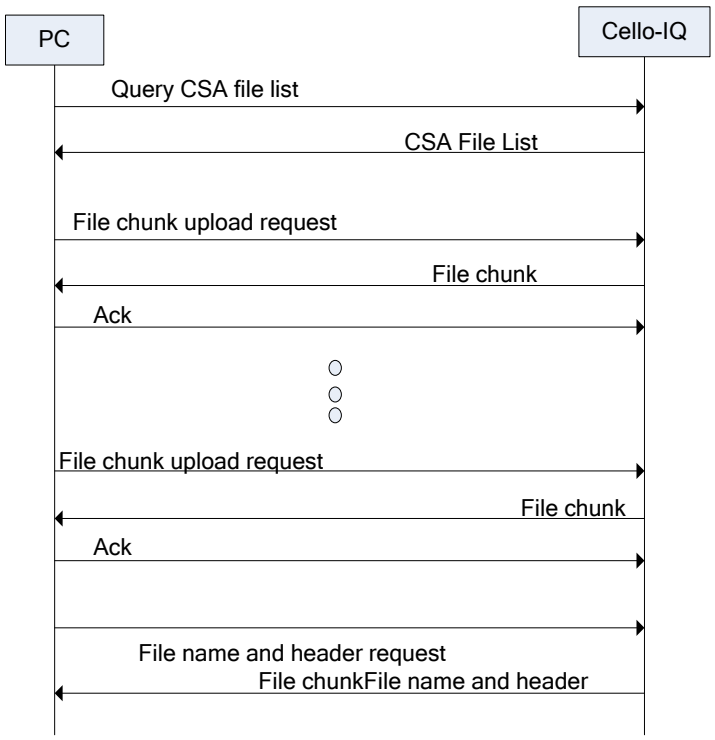
The SW will initiate the upload by requesting list of available file IDs. The command has different request for trip files and crash files.

As a response the unit will send list of all available file IDs, maximum 300 values (up to 4 chunks).

Based on the ID list, the Sw will ask chucks of data associated with the selected file ID. The Sw has to acknowledge each payload chunk.

After all the chunks are loaded the Sw will ask for the files header.

The upload shall utilize the following flowchart.



4.4.30.7 Query Trip/Crash List (Inbound)

	Command ID (Using module 50)– 45 Query Trip/Crash list
	Module length (2)
	Type of command 0 – List crash files 1 – List trip files
	Frame number

4.4.30.8 Trip/Crash List Response (Outbound)

	Command ID (Using module 50)– 45 Trip/Crash list
	Module length
	Type of command 0 – List crash files 1 – List of trip files

	Frame control Frame number / Out of total frames, Spare Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0
	ID of oldest trip
	Length of trip/crash (bytes)
	ID of oldest trip
	Length of trip/crash (bytes)

4.4.30.9 Chunk Upload Request (Inbound)

It is forbidden to upload file by non-sequential addresses (otherwise the CRC will not be calculated properly). The application is allowed to repeat the request of the same chunk, or to request the next chunk, starting from the last address of the previous chunk. In case of ivalid chunk request the unit will not answer.

	Command ID (Using module 50)– 46 Chunk Upload Request
	Module length 11
	File type (Crash 0 / Trip 1)
	Trip/Crash ID

	First address
	Length of chunk

4.4.30.10 Uploaded Chunk Response (Outbound)

	Command ID (Using module 50) (46 - Uploaded File Chunk)
	Module Length
	File type (Crash 0 / Trip 1)
	File ID
	First address
	The chunk (payload)

4.4.30.11 Uploaded Chunk Response ACK (Inbound)

Confirmation of software about correct reception of the chunk.

	Command ID (Using module 50) (47)
	Module Length 1
	ACK 0 / NACK1

4.4.30.12 Uploaded File Name and Header Request (Inbound)

Sent by the software after the file is received

	Command ID (Using module 50)– 48 Name and header request
	Module length
	File type (Crash 0 / Trip 1)
	Trip/Crash ID

4.4.30.13 Uploaded File Name and header Response (outbound)

Sent by the software after the file is received

	Command ID (Using module 50)– 48 Name and header request
	Module length
	File type (Crash 0 / Trip 1)
	File name length
	File name (ASCII), variable length
	File Header length
	File name header. Refer to File Detailed Structure, from File type to Payload (not including)

4.4.30.14 DFD: Sound Player Control Command

This module is issued by the CSA when it is required to play DFD pre-reordered message. If the command is sent while another sound string is played, the playback will be stopped and the new sound string will be played.

Byte Number	Description / Value
10	Cello ID (4 bytes)
11	
12	
13	
14	TPA Command ID – 50 dec (outbound)

Byte Number	Description / Value
15	Control byte 0 – Stop Playing 1 - Activate
16	Language (offset of sound string) : See Note Below.
17	ID of recorded sound string : See Note Below.
18	Spare

Note that for detailed definition of language and phrase IDs please refer to: **"CSA Programming Manual: Section 6.4.2: Audio Feedback messages language selection"** . If the command is sent while another sound string is played, the DFD playback will be stopped.

4.4.30.15 DFD: Response to Player Control Command

This module is issued by DFD as a reply to "Sound player control command".

Byte number	Description / Value
10	Cello ID (4 bytes)
11	
12	
13	
14	TPA Command ID – 51 dec (inbound)
15	Control byte 0 – ACK (Recorded string) 1 – NACK (Requested string does not exist) 3 - NACK (Failure)
16	Language (offset of sound string)
17	ID of playing sound string
18	Duration (how long is the selected string being played) [milliseconds]
19	
20	Spare

4.4.30.16 DFD: LEDs Control Command

This module is issued by the CSA when it is required to change LEDs activation state.

<table>
<tr><th>Byte number</th><th>Description / Value</th></tr>
<tr><td>10</td><td rowspan="4">Cello ID (4 bytes)</td></tr>
<tr><td>11</td></tr>
<tr><td>12</td></tr>
<tr><td>13</td></tr>
<tr><td>14</td><td>TPA Command ID – 52 dec (outbound)</td></tr>
<tr><td>15</td><td>Control byte
0 – Status Request (for this case bytes 16-18 are don’t care)
1 – LED control Command</td></tr>
<tr><td>16</td><td>LED Activation bitmask (1st byte). 1 – Activate LED, 0- Deactivate LED
<table>
<tr><td>Status bar Lowest (1st)</td><td>RPM</td><td>Off road</td><td>Idling</td><td>Brake</td><td>Turn</td><td>Acceleration</td><td>Speeding</td></tr>
<tr><td>LED 8</td><td>LED 7</td><td>LED 6</td><td>LED 5</td><td>LED 4</td><td>LED 3</td><td>LED 2</td><td>LED 1</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>17</td><td>LED Activation bitmask (2nd byte). 1 – Activate LED, 0- Deactivate LED
<table>
<tr><td colspan="4" rowspan="2">Reserved</td><td>Status bar Highest (5th)</td><td>Status bar 4th</td><td>Status bar 3rd</td><td>Status bar 2nd</td></tr>
<tr><td>LED 12</td><td>LED 11</td><td>LED 10</td><td>LED 9</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>18</td><td>Spare</td></tr>
</table>

4.4.30.17 DFD: Response to LEDs Control Command

This module is issued by DFD as a response to LEDs control command.

<table>
<tr><th>Byte number</th><th>Description / Value</th></tr>
<tr><td>10</td><td rowspan="4">Cello ID (4 bytes)</td></tr>
<tr><td>11</td></tr>
<tr><td>12</td></tr>
<tr><td>13</td></tr>
<tr><td>14</td><td>TPA Command ID – 53 dec (inbound)</td></tr>
</table>

<table>
<tr><th>Byte number</th><th>Description / Value</th></tr>
<tr><td>15</td><td>Control byte
0 – ACK
1 – NACK (unknown failure)</td></tr>
<tr><td>16</td><td>LED Activation bitmask (1st byte). 1 – Active LED, 0- Inactive LED
<table>
<tr><td>LED 8</td><td>LED 7</td><td>LED 6</td><td>LED 5</td><td>LED 4</td><td>LED 3</td><td>LED 2</td><td>LED 1</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>17</td><td>LED Activation bitmask (2nd byte). 1 – Activate LED, 0- Deactivate LED
<table>
<tr><td colspan="4">Reserved</td><td>LED 12</td><td>LED 11</td><td>LED 10</td><td>LED 9</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>18</td><td>Spare</td></tr>
</table>

4.4.30.18 DFD: Maintenance Command

This module is issued by CSA to activate different functions of DFD.

Byte number	Description / Value
10	Cello ID (4 bytes)
11	
12	
13	
14	TPA Command ID – 54 dec (outbound)
15	Command ID
16	Data field 1
17	Data field 2

Values

Command ID	Data field 1	Data field 2	Description / Value
0	Don’t care	Don’t care	Status request
1	Don’t care	Don’t care	Repeat self test
2	Needed volume level (0-100%)	Don’t care	Set volume level

Command ID	Data field 1	Data field 2	Description / Value
3	Needed brightness level 0 – day mode 1 – night mode	Don't care	Set brightness level
4	Don't care	Don't care	Reset
5	Sound feedback 0 – disable 1 - enable	Visual feedback 0 – disable 1 - enable	Disable/Enable self-activated sound/visual feedbacks

4.4.30.19 DFD: Status (also response to Maintenance Command)

This module is issued by DFD as a response to a command AND upon power up, after the self-test.

Byte number	Description / Value
10	Cello ID (4 bytes)
11	
12	
13	
14	TPA Command ID – 55 dec (inbound)
15	Control byte 0 – ACK to command 1 – Self initiated status update
16	Self-Test result 0 – OK 1 – Failure 1 … 255 – Failure 255
17	Self-initiated Feedback status -Sound 0 – Disabled 1 - Enabled
18	Self-initiated Feedback status -LEDs 0 – Disabled 1 - Enabled

Byte number	Description / Value
19	Volume level %
20	Brightness level 0 – day mode 1 – night mode
21	Spare (sent as zero)
22	DFD firmware Version
23	
24	
25	
26	SD Card file version
27	
28	
29	
30-39	Spare

4.4.30.20 DFD: Serial Number Read/Set Command

This module is issued by CSA to set an ID of DFD.

Byte number	Description / Value
10	Cello ID (4 bytes)
11	
12	
13	
14	TPA Command ID – 56 dec (inbound)
15	Control byte 0 - Read/1 – Set In case of 0 bytes 16-19 don’t care.
16	DFD ID (LSB)
17	DFD ID (2nd byte)

Byte number	Description / Value
18	DFD ID (3rd byte)
19	DFD ID (MSB)

4.4.30.21 DFD: Response to DFD Serial Number Read/Set Command

This module is issued by DFD as a response to ID request.

Byte number	Description / Value
10	Cello ID (4 bytes)
11	
12	
13	
14	TPA Command ID – 57 dec (outbound)
15	Spare
16	DFD ID (LSB)
17	DFD ID (2nd byte)
18	DFD ID (3rd byte)
19	DFD ID (MSB)

4.4.30.22 DFD: Default Audio String Read/Set Command

This module is issued by CSA to set an ID of DFD.

Byte number	Description / Value
10	Cello ID (4 bytes)
11	
12	
13	
14	TPA Command ID – 58 dec (inbound)
15	Control byte 0 - Read/1 – Set In case of 0 bytes 16-17 don't care.
16	String number (upon no communication) Default – 03.

Byte number	Description / Value
17	Language (Default 1 – English)

4.4.30.23 DFD: Response to Default Audio String Read/Set Command

This module is issued by DFD as a response to ID request.

Byte number	Description / Value
10	Cello ID (4 bytes)
11	
12	
13	
14	TPA Command ID – 59 dec (outbound)
15	Spare
16	String number (upon no communication) Default – 03.
17	Language (Default 1 – English)

4.4.31 *PointerCept Base Station – Status Request/Power Control (Type 0x40)*

This message is sent from the server (or mobile app) to trigger a Beacon Detected/Status message from PointerCept base station, or other certain base station action (Power, output activation, etc.).

Byte	Data
1	'M'
2	'2'
3	'C'
4	0x40
5	Unit ID (of the base station MCU)
6	
7	
8	
9	Command Code 0 - Spare. 1 - Status Request – reply with Status Message

	2 - Beacon Detect Request - reply with last Beacon Status message for each receiver (for all activated receivers) 3 - Activate output 4 - Deactivate output 5 - Full System Reset (including Cello unit) 6 - Activate in Full Power mode - regardless of ignition status (not in use) 7 - Activate in Average Hibernation mode - regardless of ignition status (not in use) 8 - Activate in Full Hibernation mode (not in use) 9 - Base station turn off - deadly, including sending dedicated Power status message
10	Spare
11	Checksum

4.4.32 *PointerCept Base Station – Chase Start/End/Location Update (Type 0x41)*

This message is sent from the server (or mobile app) to switch PointerCept base station into/out of Chase mode, and as a response to Location Request from the mobile app.

Byte	Data
1	'M'
2	'2'
3	'C'
4	0x41
5	Unit ID (of the base station MCU)
6	
7	
8	
9	End Unit Under Chase ID
10	
11	
12	
13	Group 0-5
14	Command Code 0 – Chase start request from server If received from server it is routed to the mobile app (via BT) for user chasing approval 1 – Reserved

	2 – Location update Reply upon Location request by the user, routed to mobile app (via BT) 3 – Start chase mode The base station MCU configures all 6 receivers with the required specific group for enhanced chasing sensitivity. 4 – End chase mode The base station MCU returns back to PL defined operation mode. The Group field (byte 13) is ignored. For all of the above if BS starts to operate in chasing mode (via BT request or server request) byte 11 in the Status message (0x41) generated by BS is changed accordingly.
15	Last GPS Fix
16	
17	Longitude
18	
19	
20	
21	Latitude
22	
23	
24	
25	Location (0 – GPS based, 1 – RF based)
26	Speed direction: Direction (angle) of the speed vector mapped
27	Spare
28	Checksum

If the message was received with chase start request from server (byte 14 = 0), the PointerCept base station forwards the Chase Start/End/Location Update message (with message type 0x42) to the mobile app for user approval.

If the message was received with location update from server (byte 14 = 2), the PointerCept base station forwards the Chase Start/End/Location Update message (with message type 0x42) to the mobile app.

4.4.33 *Nano Status Interrogation (Type 0x37)*

This command is for both inbound and outbound messages.

Byte	Data
1	'M'

2	'2'
3	'C'
4	0x37
5	Reserved (0x000000)
6	
7	
8	N-Length of data forwarded (from byte 10 to N+9) (Two bytes)
9	
10	Payload Data (N bytes). This shall contain the type-11 modules.
11	
..	
..	
N+10	Checksum – Legacy M2C terminating Checksum

Currently, the only supported Type-11 command is this:

Byte	Description	Containing
0	Module Name 32 = General Command	32
1	Length of module (16 bits)	6
2		
3	Number of Command entries sent by this module (8 bits)	1
4	Command ID	259 = Nano: send status of the designated source
5		
6	Length of entry data	2
7	Source of measurement (single only)	Sources enumeration according to this definition: 0x00÷0x0F – MultiSense unit (index number in the system) 0xFD – Nano
8	Reserved	0

The response is a standard type-11 packet with modules 8, 6, 7 and 42 in case of Nano and with 8, 6, 7, 42 and 28 in case of MultiSense. Where the module 28 is the "provisioning" message containing the entire status of the MultiSense.

4.4.34 *Nano and CelloTrack-4: Request to send a distress event to server (Type 0x39)*

This command is inbound (only) to the unit, requesting the unit to send a type-0 distress event with location to the server.

This is used as kind of ping to the servers.

Byte	Data
1	'M'
2	'2'
3	'C'
4	0x39
5	Reserved (0x000000)
6	
7	
8	Server type 0= Operational 1-255 = Reserved
9	Spare
10	
11	
12	Checksum – Legacy M2C terminating Checksum

4.4.35 *Nano and CelloTrack-4: RTC push (Type 0x3A)*

This command is inbound (only) to the unit, pushing the RTC from a PC (at production line for example).

A normal ACK (as detailed in the Serial protocol doc) will be sent back from to unit to the requester.

Byte	Data
1	'M'
2	'2'
3	'C'
4	0x3A
5	Reserved (0x00000000)
6	
7	
8	
9	RTC stamp (UTC)- Seconds
10	RTC stamp (UTC)- Minutes
11	RTC stamp (UTC)- Hours
12	RTC stamp (UTC)- Day

13	RTC stamp (UTC)– Month
14	RTC stamp (UTC)– Year (-2000)
15	Spare
16	
17	
18	Checksum – Legacy M2C terminating Checksum

4.4.36 *Unit inactivation*

When CelloTrack Gen-4 family unit receives this command, the unit will go to inactive state (shut itself down).

Byte	Data
1	'M'
2	'2'
3	'C'
4	Message type - 0x47
5	Unit ID
6	
7	
8	
9	Spare
10	Checksum

4.4.37 *Pass through (Type 0x46)*

Byte	Data
1	'M'
2	'2'
3	'C'
4	0x46
5	Device enumeration: 0 – Feature disable

	1 – Tunnel the **Modem** communication to the serial port 2 – Tunnel the **GPS** communication to the serial port 3 - Tunnel the **Satellite** communication to the serial port 4 - Tunnel the **BT** communication to the serial port 5-255 - Reserved
6	Pass through timeout
7	0 - Pass-through without a timeout 1- 65536(sec)
8	Power Control 0 - None 1 - Applicative turn ON 2 - Applicative turn OFF 3 - Power ON 4 - Power OFF 5-255 Reserved
9	Reserved
10	
11	
12	
13	
14	
15	Checksum

4.4.38 *Modem FOTA Command (Type 0x50)*

This packet enables activation of Modem FOTA upgrade via FTP server (which hosts a delta file received from advance). It contains the details of the desired FTP server, directory, and upgrade file.

Upon receiving this command, the unit will send ACK to the requesting channel (packet 0x00).

After completing the upgrade process, the unit will reply on this command with Modem FOTA Response (packet 0x50).

Byte	Data
1	'M'
2	'2'
3	'C'
4	0x50
5	Unit ID

<table>
<tr><td>6</td><td rowspan="3"></td></tr>
<tr><td>7</td></tr>
<tr><td>8</td></tr>
<tr><td>9</td><td>Control Byte
<table>
<tr><td>Spare</td><td>Spare</td><td>Spare</td><td>Spare</td><td>Spare</td><td>Spare</td><td>Spare</td><td>Spare</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>10</td><td>Spare</td></tr>
<tr><td>11</td><td>Spare</td></tr>
<tr><td>12</td><td>FTP Server IP Address Byte 0 (LSB)</td></tr>
<tr><td>13</td><td>FTP Server IP Address Byte 1</td></tr>
<tr><td>14</td><td>FTP Server IP Address Byte 2</td></tr>
<tr><td>15</td><td>FTP Server IP Address Byte 3 (MSB)</td></tr>
<tr><td>16</td><td rowspan="2">FTP Server IP Port (0-65535)</td></tr>
<tr><td>17</td></tr>
<tr><td>18</td><td>Spare</td></tr>
<tr><td>19</td><td>Spare</td></tr>
<tr><td>20-49</td><td>FTP Server Username (ASCII String, length 30, zero padded)</td></tr>
<tr><td>50-79</td><td>FTP Server Password (ASCII String, length 30, zero padded)</td></tr>
<tr><td>80</td><td>Spare</td></tr>
<tr><td>81</td><td>Spare</td></tr>
<tr><td>82-151</td><td>Full Path (ASCII String, length 70, zero padded)</td></tr>
<tr><td>152</td><td>Spare</td></tr>
<tr><td>153</td><td>Spare</td></tr>
<tr><td>154-203</td><td>Full File Name (ASCII String, length 50, zero padded)</td></tr>
<tr><td>204</td><td>Checksum</td></tr>
</table>

5 Examples and Final Notes

5.1 Final Notes

Regarding configuration memory reading and writing – should the end address of a region (end = start+(length-1)) be more than highest possible address (1024 in oldest units, 2048 in Compact and 4096 in CelloEDGE), a wrap-around takes place once the unit attempts to advance from highest address, and therefore the address is set to zero.

5.2 Examples

5.2.1 *Status/information Request Packet*

5.2.1.1 3-Wire Serial Interface

The command: 4D 32 43 00 00

Unit response: 4D 32 43 02 37 00 00 00 04 17 04 0E 00 00 00 00 66

5.2.1.2 4-Wire Interface (obsolete)

RTS line raises

Unit responds by: 4F 4B

The command: 00 00

Unit response: 4D 32 43 02 37 00 00 00 04 17 04 0E 00 00 00 00 66

5.2.2 *Forward Data to Wireless Channel Command*

5.2.2.1 3-Wire Interface

The command: **4D 32 43** 08 05 48 65 6C 6C 6F 01

ACK from the unit: 4D 32 43 00 37 00 00 00 37

5.2.2.2 4-Wire Interface (obsolete)

RTS line raises

Unit responds by: 4F 4B

The command: 08 05 48 65 6C 6C 6F 01

ACK from the unit: 4D 32 43 00 37 00 00 00 37