# Cellocator OTA Firmware Upgrade


Proprietary and Confidential

Version 1.1

Revised and Updated: May 15, 2013


Pointer Telocation Ltd. 14 Hamelacha St., Rosh Ha'ayin 48091, Israel • Tel: 972-3-5723111 • Fax: 972-3-5723100 • www.pointer.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 Pointer Telocation Ltd. may be hazardous and/or may cause damage to the product and should not, in any circumstances, be used or combined with the product.

## General

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

## Warranty Exceptions and Disclaimers

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

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

## Intellectual Property

# Table of Contents

# 1 Introduction

## 1.1 Overview

This document describes the OTA firmware upgrade process, which includes the following capabilities:

- The FW update (Reflash) ability of the Cellocator unit allows the updating the FW section of the unit's memory without any external equipment.
- The firmware upgrade can be used via a wire connection (to the unit's serial interface) or via the wireless channel of the unit (OTA).
- The firmware upgrade allows updating installed units without necessitating the removal of the units for the updating process.
- This upgrade is based on a lightweight scripting engine which allows for both minor and major patching of the firmware. This makes this ability cost-efficient, allowing for small patches of firmware to be made with minimal traffic and time.

## 1.2 Revision History

| Version | Date | Description |
|---|---|---|
| 1.0 | 14/5/06 | First draft. |
| 1.1 | 15/5/2013 | Improving look and feel<br>Aligning with updated products |

## 1.3 Abbreviations

CSF – Cellocator script file

OTA – over the air

Chunk – group of bytes which are a portion of a file.

# 2 FW Update (Reflash) Description

## 2.1 FW Update (Reflash) Process overview

1. A CSF file is provided by Cellocator.
2. The file is downloaded to the desired end-units using Cellocator's or customer's software.
3. Upon termination of download and verification of the file by the unit, the unit resets itself, performs the re-flashing and resets itself again with the updated firmware.

## 2.2 CSF File Structure

### 2.2.1 *File headers:*

1. 4 bytes of text header, values – ASCII for "FCSR".
2. 2 bytes of file type (1 byte major, 1 byte minor).

### 2.2.2 *Target platform identifiers:*

1. This part is a list of identifier records for appropriate target platforms which are suitable for updating with this file.
2. The first byte of this part is the "count byte". It contains the amount of records in the list.
3. Immediately following the count byte are the actual target platform records. The records are saved successively with no padding or any data between the records. The amount of records is as defined in the count byte.
4. A compatible target platform record is a record whose identifiers match those in the platform manifest of the target unit. Some identifiers of the target record may also be zero, in which case they represent "don't-care" fields, that is – fields which should not be tested for matching the target unit's manifest.
5. Target Platform Identifiers Record structure:
   - 1 byte target processor family identifier.
   - 1 byte - target hardware interface and peripherals identifier.
   - 2 bytes - minimal size of program memory (in 1024 words or 2048 bytes units).
   - 2 bytes - maximal size of program memory (in 1024 words or 2048 bytes units).
   - 2 bytes - minimal size of volatile memory (amount of actually available general-purpose bytes), divided by 128 bytes and rounded up/down to closest integer.
   - 2 bytes - maximal size of volatile memory (amount of actually available general-purpose bytes), divided by 128 bytes and rounded up/down to closest integer.
   - 2 bytes - minimal size of internal (on processor) non-volatile memory (e.g. Flash or EEPROM), divided by 128 bytes and rounded up/down to closest integer.
   - 2 bytes - maximal size of internal (on processor) non-volatile memory (e.g. Flash or EEPROM), divided by 128 bytes and rounded up/down to closest integer.
   - 2 bytes - minimal size of external (on board) non-volatile memory (in 1024 words or 2048 bytes units).

- 2 bytes - maximal size of (on board) external non-volatile memory (in 1024 words or 2048 bytes units).
- 1 byte - external non-volatile memory type.
- 1 byte - hardware version.
- 2 bytes - reprogramming facility identifier.
- 2 bytes - script language version.

### 2.2.3 *The Firmware (encoded):*

1. 4 bytes - length of this part (the Reflash script), excluding this length field.
2. A variable - amount of bytes (as specified in the previous length field) containing the actual script commands and data to be updated. This part is the data that should be transmitted to the end unit being updated.

### 2.2.4 *Meta data:*

1. CRC32 of the firmware part only.
2. File number.
3. New firmware identifier (2 bytes, unique value for each firmware).

### 2.2.5 *File authentication hash (MD5)*

It contains the MD5 hash of the whole file.

## 2.3 CSF File Detailed Download Process

1. The end units provide the following facilities via the communication interfaces (wireless and wire):
   a) Manifestation of their firmware platform.
   b) Establishment of file download mode.
   c) File portion (chunk) reception and acknowledgement.
   d) Completion (or abortion) of file download.
2. The download of the CSF file takes place in the following manner:
   a) The updating host begins communicating with the end-unit by inquiring for its firmware platform information.
   b) The end-unit manifests its firmware and hardware platform.
   c) The host examines the platform identifications and its own database, and accordingly decides whether to proceed with the operation or to abort.
   d) The host selects an appropriate target platform identifiers record (as there may be more than one in a file) from an appropriate CSF file in its database and attempts to establish file download mode with the unit, sending the file's target platform identifiers record.
   e) The unit examines the target platform identifiers. If they are compatible with its own then the unit ACKs and enters file download mode. Otherwise, it NACKs.
   f) The host sends the file's Reflash script part in chunks, while keeping track of which chunks have been ACKed and which chunks have not.

g) The unit ACKs the received file chunks.

h) After all chunks have been sent and ACKed, the host notifies the completion of file download and transmits the file meta-data.

i) The unit processes the file and the meta-data to authenticate the data. If everything is in order, the unit ACKs and proceeds to reflash the file.

## 2.4 Wireless Protocol Appendix for FW Updating

### 2.4.1 *Inbound Messages*

1. Firmware Platform Manifest Request
   - The request causes the unit to transmit in return the "Firmware Platform Manifest" message.
   - This request is sent via the "modular request" message (code 9), with sub-data type code 0x01.
2. File Download Control
   - The command is used to establish, complete or abort file download mode, as well as actually transfer file chunks.
   - This command uses message type ID 0x0A, structured as follows:
     1. 14 bytes standard incoming messages header (system code, message type=0x0A, target unit ID, numerator, auth code).
     2. 1 byte - total byte length of message-specific data (applying to all of the following bytes).
     3. 1 byte - desired download action: value "0" for abortion of any process underway; value "1" for establishing file download process; value "2" for completion of file download process; value "3" for transfer of file chunk.
     4. Associated data:
        - Process abortion (0) – no further data.
        - Process establishing (1) – 24 bytes of target platform identifiers – a simple verbatim transfer of the relevant target platform IDs record from the file.
        - Process completion (2) – file meta-data, all of which is contained in the file from which the downloaded data originated:
          - ✓ Total length of the reflashing script (4 bytes).
          - ✓ Reprogramming script hash (CRC32).
          - ✓ File number (16 bytes).
          - ✓ New firmware ID (2 bytes).
        - File chunk transfer (3) – 3 bytes chunk's base offset, followed by a chunk of the file's data. The actual amount of data bytes is derived from the total length of the message (message's data length minus 4).

### 2.4.2 *Outbound Messages*

1. Master ACK/NACK message:
   - This message notifies the proper/improper execution of a command.
   - This command is a new message type with ID 0x0A, structured as follows:

- 12 bytes - standard header (system code, message type = 0x0A, target unit ID, comm. control, numerator)
- 1 byte - announcement type ("1"=ACK, "2"=NACK).
- 1 byte - process-specific error/success code.
- 6 bytes - spare.

**Process-specific error/success code description:**

| Description | Value |
|---|---|
| No errors, read/program/verify/everything OK. | 0 |
| Download commands received but download mode isn't established. | 2 |
| Download commands received after completion was already performed. | 3 |
| Incompatible scripting/reflashing version. | 0x10 |
| Establishment failed due to incompatible platform. | 0x20 |
| CRC-32 test failed. | 0x40 |
| External EEPROM verify fail -- either after chunk save command or after trying to save some parameters. | 0xE0 |
| Internal EEPROM programming error (auto-verify fail). | 0xF0 |
| Busy, cannot perform the action at the moment. | 0xF1 |

2. Firmware Platform Manifest:
   - This message presents the firmware platform manifest of the unit.
   - The message is sent via the "modular data" message (code 9), with the following data:
     1. 1 byte data type, value 0x01.
     2. 1 byte data length, value 0x10.
     3. 1 byte processor family identifier.
     4. 1 byte hardware interface and peripherals identifier.
     5. 2 bytes size of program memory (in 1024 words units).
     6. 2 bytes size of volatile memory (divided by 128 bytes and rounded up/down to closest integer).
     7. 2 bytes size of internal non-volatile memory (divided by 128 bytes and rounded up/down to closest integer).
     8. 2 bytes size of external non-volatile memory (in 1024 words units).
     9. 1 byte external non-volatile memory type.
     10. 1 byte hardware version (same as in status message).
     11. 2 bytes reprogramming facility identifier. This document defines facility type 1 (one). First byte is therefore 0x01, the second is zero.
     12. 2 bytes script language version – the version of script language supported by the unit (1 byte major, 1 byte minor).

13. 2 bytes current firmware ID. Note that this is in fact not a descriptor of the firmware platform per se, but rather a descriptor of the actual firmware running on the platform. However, it is a valuable field when a Reflash is considered.

## 2.5 Reflashing Event

After the end of the Reflashing process the unit generates and stores a Reflash Confirmation event with the transmission reasons described in the table below:

| Reason | Description |
|---|---|
| 0xF7 | Self Programming Success |
| 0xD0 | Self Programming Aborted - illegal firmware file header detected |
| 0xD1 | Self Programming Aborted - unsupported self programming scenario |
| 0xD2 | Self Programming Aborted - unsupported command detected |
| 0xD3 | Self Programming Aborted - illegal data detected in one of the commands |
| 0xF8 | Self Programming Aborted - communication with external memory failed |

The event will be delivered after re-establishing communication with the server.