Cellocator Cello Family – Squarell 8000

Integration Manual

Proprietary and Confidential

Version 1.0

Revised and Updated: May 22, 2013

ION LTD. 14 HAMELACHA ST., ROSH HA'AYIN 48091, ISRAEL • TEL: 972-3-5723111 • FAX: 972-3-5723100 • www.pointer.com

Copyright © 2013 by Pointer Telocation, Ltd.

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.

Copyright © 2013 by Pointer Telocation, Ltd.

Intellectual Property

Copyright in and to this document is owned solely by Pointer Telocation Ltd. Nothing in this document shall be construed as granting you any license to any intellectual property rights subsisting in or related to the subject matter of this document including, without limitation, patents, patent applications, trademarks, copyrights or other intellectual property rights, all of which remain the sole property of Pointer Telocation Ltd. Subject to applicable copyright law, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording or otherwise), or for any purpose, without the express written permission of Pointer Telocation Ltd.

© Copyright 2013. All rights reserved.

Copyright © 2013 by Pointer Telocation, Ltd.

Table of Contents

Copyright © 2013 by Pointer Telocation, Ltd.

1 Introduction

This document describes the recommended configuration and serial port (RS232) integration process for Cellocator Cello family devices connected with Squarell's 8000 series devices. This document also provides a complete description of all the supported messages and parameters by Squarell's 8000 series.

Using the recommended RS232 connectivity and the advanced transparent protocol provided by Cellocator devices, a fully functioning fleet management solution with in-vehicle data connectivity for remote diagnostics, vehicle performance and driver behaviour can be fulfilled within minutes.

IMPORTANT: Not all parameters from the list of Squarell's supported parameters may be available. This availability of the data depends on the brand, model, year and configuration of the vehicle.

1.1 Revision History

Version	Date	Description
121026	26-10-2013	Pointer Cellocator Protocol described
130411	11-04-2013	Screenshots and configuration added

Copyright © 2013 by Pointer Telocation, Ltd.

2 Integration steps

2.1 Standard RS-232 settings

The Squarell 8000 series Device translates the internal CANbus (J1939 or proprietary), J1708 (J1922 or J1587) and K-Line (DTCO) protocols into RS232 messages with following port settings. These settings should be also used by Cellocator Cello device:

The RS232 port set-up for ASCII with the Pointer Cellocator:

- Bits per second 9600
- Data bits 8
- Parity No
- Stop bits 1
- Flow Control (Handshaking) No

2.2 Remote Tachograph download settings

The Squarell 8000 series Device will switch to the following settings when Remote Tacho Download is enabled (Starting from DCF versions 1-10-2012).

The RS232 port set-up for ASCII ESCAPED:

- Bits per second 9600
- Data bits 8
- Parity No
- Stop bits 1
- Flow Control (Handshaking) No
- ACCM 0

If ASCII ESCAPED is used, a checksum according to HDLC/PPP is added to the message before the <CR><LF>.

To accommodate easy testing the checksum 9999 is always passed (Squarell will accept all messages with this checksum).

Copyright © 2013 by Pointer Telocation, Ltd.

2.3 Connect Cellocator Cello family with Squarell Devices

There is minimal connection between the Squarell device and the Telematics device:

- RX signal wire
- TX signal wire
- Ground signal wires

2.3.1 *Connections on Cellocator Cello Family Devices*

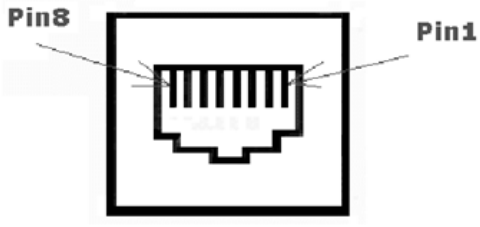
The Cellocator device has a 2 wire RS232 port. The pinning of the Cellocator is as per the following:

- Cello Pin 3: GND
- Cello Pin 12: RS232 Tx
- Cello Pin 13: RS232 Rx

Connection using an RJ45 female connector:

Typically Cellocator's full harness (item # 711-00248/A) supports an RJ45 female connector for External Data, allowing external device communication to the Cello via its RS232 interface (pins 12 and 13).

The RJ45 connector is illustrated in the following figure:



Serial Port Adaptor Connector – Front View

Copyright © 2013 by Pointer Telocation, Ltd.

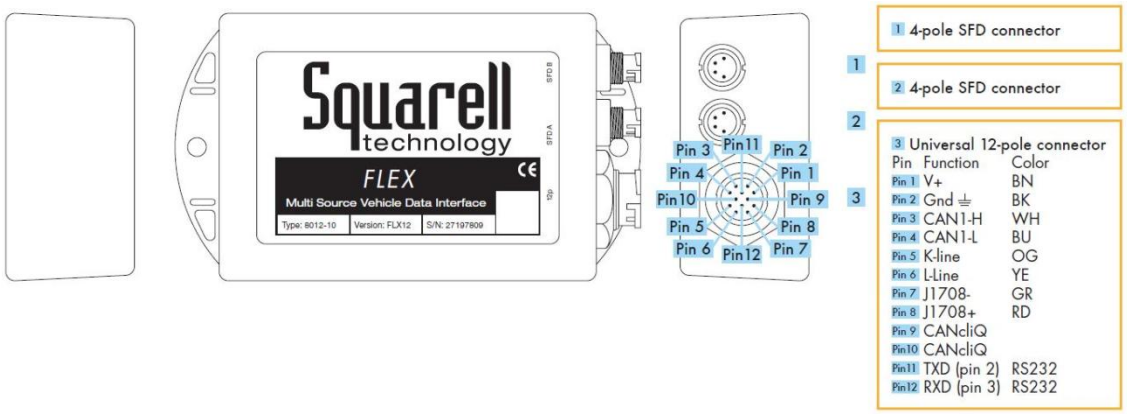
When a RJ45 connector supported harness is being used, the pinning is per the following:

- Cello RJ45 Pin 1: GND
- Cello RJ45 Pin 3: RS232 Tx
- Cello RJ45 Pin 4: RS232 Rx

For evaluation purposes, one may use Cellocator's RJ45 to DB9 adapter [CN 711-0078] for easier connection.

2.3.2 *Connection with Squarell Flex Device*

The female serial port adaptor connector pin out of the Squarell Flex device is:



Flex Connections:

- Dsub9 Pin2 (RS232 TX) to Cello RJ45 Pin 3
- Dsub9 Pin3 (RS232 RX) to Cello RJ45 Pin 4
- Dsub9 Pin5 (Ground) to Cello RJ45 Pin 1

Copyright © 2013 by Pointer Telocation, Ltd.

2.4 Setting up the Cellocator Cello using the Communication Center

Cellocator's Cello devices require specific programming settings in order to route the messages in good order to the server. For this purpose the settings of message type 7 (transparent forwarding of data from Squarell to Server) and message type 5 (transparent forwarding of commands from the server to the Squarell device) are used for end to end communication fulfilment.

Communication Center software can be used to easily program the Cellocator unit, monitor messages and send commands from the device during the integration evaluation process. Please refer to Cellocator's Evaluation Manual for a detailed description of the evaluation environment establishment and main communication (Serial port attributes) and network setup parameters.

2.4.1 *Programming bit name: "Enable Transparent Mode (of COM port)"*

Address: 0 bit 7

Description: Set to '1' in order to enable transparent mode. Once transparent mode is enabled (this bit is set to 1), it may be activated/deactivated using the trigger on "door" input. When this input is in its "active" state, transparent mode is activated; when "door" is deactivated – the COM port is in CSP (Cellocator Serial Protocol) mode.

Note that the "active" state may be controlled using the programmable inputs inversion masks. In addition, any programmed events for the "door" input will be generated as usual. To avoid this, disable events associated with this input in the programming.

NOTE: Transparent mode can be activated also by OTA command (from FW31c).

---

a. Upon reception of the command, the unit immediately enters transparent mode irrespective of the status of the corresponding bit in the EEPROM *(from FW31c)*.

b. If transparent mode started as a result of OTA command, Door input is NOT used by the unit for transparent mode control.

   The door input is controlling the start and stop of transparent mode only if a corresponding bit in EEPROM is enabled. If the same bit is disabled, the transparent mode can still be started / stopped, but only by OTA command.

c. If transparent mode started as a result of OTA command, the unit will leave transparent mode only upon reception of a corresponding OTA command.

When transparent wire mode is active:

- All Cellocator wire protocol processing is ceased. This means wire commands, such as programming, mode change, etc are ignored.
- Any data received in the port is accumulated and forwarded to the wireless channel in Cellocator protocol data forwarding packets (only type 7 is discussed in this manual), containing the data as it was received, without further processing or filtering.

Copyright © 2013 by Pointer Telocation, Ltd.

Technical information

In transparent wire mode, the received serial data is packetized, so it may be sent in packets. The following is an explanation of how the data is packetized:

As data is received, it is accumulated in packets. The accumulating packet is "closed" and queued for transmission once one of the following occurs:

- *512 bytes were accumulated in the packet (this is therefore the practical maximum packet size although the Cello device will keep accumulating further incoming information even after closing and queuing the first 512 bytes).*
- *500msec have passed since the reception of the first byte in the packet.*
- *300msec have passed since the last received byte.*

Once a packet is queued for transmission, the unit will attempt to reliably forward it via the wireless channel (if available and enabled in programming for the existing network conditions) or parse and log depending on the configuration.

2.4.2 *Programming bit name: "Forward data as a Container"*

Address: 285, bit 6

Description: If this bit is enabled ('1' – recommended), once the forwarded data packet is received in a RAM buffer, the payload is escorted by additional fields and wrapped in a structure called "the container". The container is a data structure, created by the Cello device in its RAM buffer upon reception of the data from the serial port.

The forwarded payload is escorted by 48 bytes of fleet management data (attached to the last byte of payload) and the total length of the payload and FM data is transmitted.

Every container is assigned with 7 bits numerator (increased every packet reception from COM port), used in the fragmentation process and reported with the container to the server side. The indication bit of report Forwarded Message Code byte in this case will contain '1' – A Container.

Forwarded Message Code
A sequential 7 bits ID of the container + indication bit (Single byte)

The Container							
Length of container starting from byte 3 (2 bytes)		The payload of forwarded data X			48 bytes of fleet management data		
Byte 1	Byte 2	Byte 3			Byte 3 + X		Byte 3+ X+48

If this bit is disabled ('0'), the payload is forwarded as received from the serial port, without any additional wrapping. The indication bit of report Forwarded Message Code byte in this case will contain '0'.

2.4.3 *Programming bit name: "Enable Data forwarding through log"*

Address: 285 bit 7.

Copyright © 2013 by Pointer Telocation, Ltd.

Description: This bit should be set (1) in order to guarantee safe delivery of the forwarded data. In this case, the container (or the payload to forward, as per the programming in address 285, bit 6) is fragmented into chunks of 54 bytes long (last one is zero-padded). The chunks are enumerated and added with the fragmentation control fields, stored in an events log memory and then uploaded as message type 7. Message type 7 is similar to other logged messages in Cellocator protocols (like message type '0' or '9') in the following ways:

- Continues the Message Numerator used by other logged messages.
- Requires acknowledge from the server (Message type 4) in order to erase the specific message from the log.
- The message 7 utilizes the same retransmission algorithms as other logged message types.

Copyright © 2013 by Pointer Telocation, Ltd.

2.5 Functionality validation

2.5.1 *Monitor FMS message from Squarell device*

Once the Cellocator device is set to the right configuration and connected through the RS232 to the Squarell device, the Communication Center can be used to monitor incoming messages that are sent by the Squarell device and forwarded by the Cellocator unit.

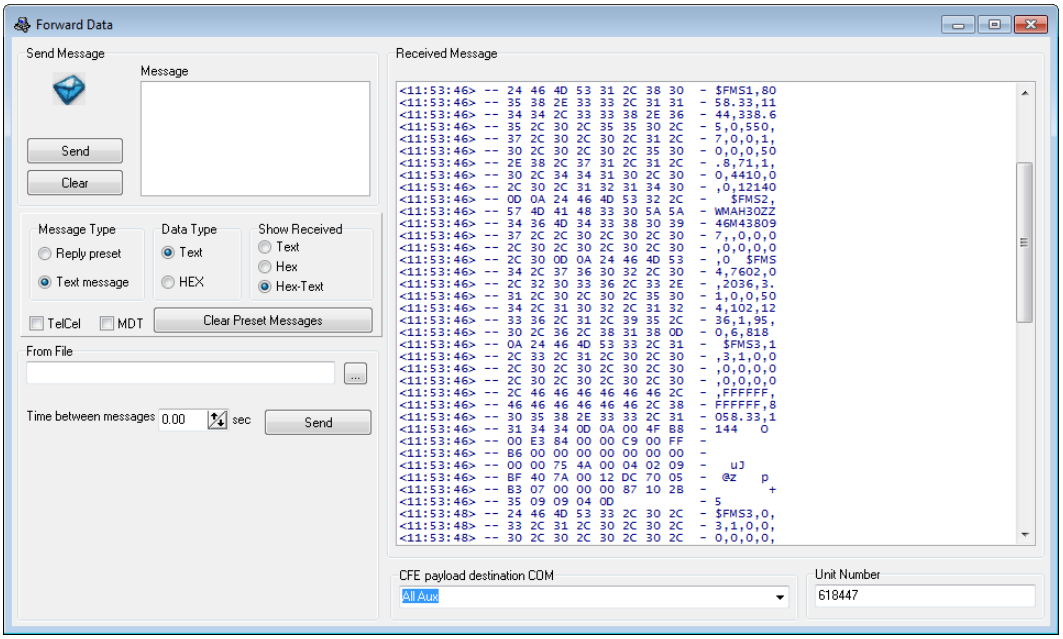
Important: Before being able to monitor the incoming messages, the Squarell device needs to be:

- Connected to the Cellocator device.
- Configured with the Cellocator specific configuration (DCF) file.
- Reading vehicle data (from a real life vehicle or a simulation setup). This setup can be found in the Squarell Telematics Development kit.
- Connecting both power and ignition to a power supply, to prevent the device going in hibernation mode.

The messages forwarded from Squarell unit to the Communication Center can be monitored in the 'Forward Data' screen. To access this screen, select the **Forward Data/MDT** checkbox on the Forward Data tab.

Forward Data Tab

The Forward Data window is displayed, as shown below.



Copyright © 2013 by Pointer Telocation, Ltd.

Forward Data Window

Note that it is possible that no data has been sent yet. This is because the FMS messages will only be triggered on an event (please refer to the Squarell RS232 protocol description further on in this document).

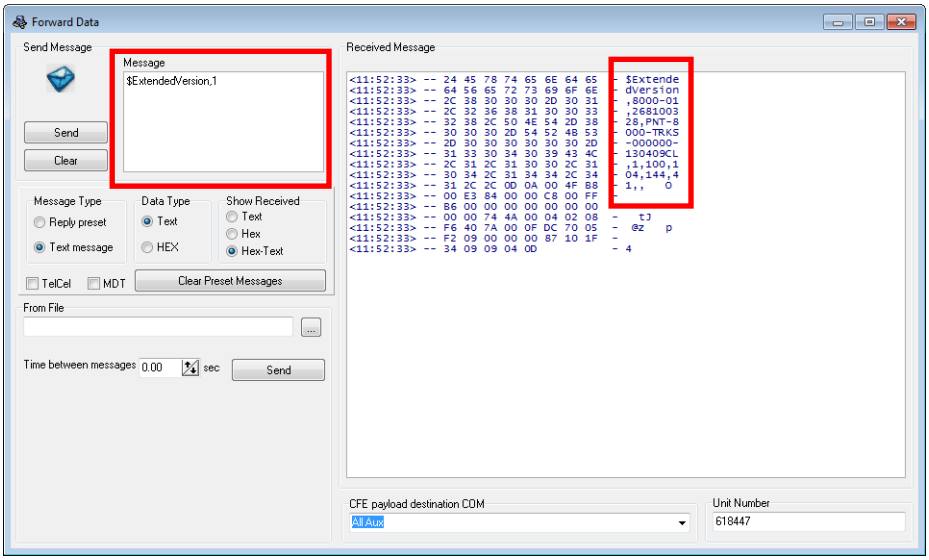
It is possible to request the FMS messages by sending a command to the Squarell device via the Cellocator unit. This is explained in the next section.

2.5.2 *Sending commands to the Squarell device using the Forward Data window*

The Forward Data window can also be used to send commands to the Squarell through the Cellocator unit. For all implemented commands, refer to the Squarell RS232 protocol description later in this guide.

While in the Forward Data window, you can see the message field on the top left of the screen. In this field you can key in a command from the RS232 protocol. In this example we've used a FMS1 request ($ExtendedVersion,1<CR><LF>).

Important: To work correctly, the 'carriage return' and 'line feed' must be replaced by 2x <ENTER>.



After clicking the Send button, the unit should respond with the message above (including device type, serial number, DCF version, firmware version, etc).

Copyright © 2013 by Pointer Telocation, Ltd.

2.6 Server side routines for handling Squarell data and data requests to the Squarell unit

To process the data send from the Squarell 8000 series device, the following aspects need to be taken into consideration:

- Implement parsing type 7 messages with the formats described below (refer to 2.6.1).
- Create database structures to process the parsed data, create business reports and functions (no further description).
- Implement the requests and setting of thresholds as type 5 messages with the formats described below (refer to 2.6.3).
- Implement device health and configuration application for settings and thresholds (refer to 2.6.4).

2.6.1 *Full parsing of OTA Msg. 7 (Logged)*

<table>
<tr><th>Byte no.</th><th>Description</th><th>Containing</th></tr>
<tr><td>1</td><td>System code, byte 1</td><td>ASCII “M”</td></tr>
<tr><td>2</td><td>System code, byte 2</td><td>ASCII “C”</td></tr>
<tr><td>3</td><td>System code, byte 3</td><td>ASCII “G”</td></tr>
<tr><td>4</td><td>System code, byte 4</td><td>ASCII “P”</td></tr>
<tr><td>5</td><td>Message type</td><td>7</td></tr>
<tr><td>6
7
8
9</td><td>Unit’s ID
(total 32 bits)</td><td></td></tr>
<tr><td>10
11</td><td>Communication Control field (same as in MSG Type 0)</td><td></td></tr>
<tr><td>12♠</td><td>Message Numerator</td><td>Sequential numerator of messages, used by ACK</td></tr>
<tr><td>13♠</td><td colspan="2">Static byte containing value 0x07</td></tr>
<tr><td>14♠</td><td colspan="2">Forwarded Message Code
Sequential 7 bits ID of the container+ container indication bit (MSB)
Assigned for each container
<table>
<tr><td>Container (1) / Simple payload – 0</td><td colspan="7">Sequential 7 bits ID of the container</td></tr>
<tr><td>Bit 7</td><td>Bit 6</td><td>Bit 6</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>

Copyright © 2013 by Pointer Telocation, Ltd.

Byte no.	Description	Containing
15♠	Fragment Control Byte First Frame | Last Frame | Fragment No (starting from 1) Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0	
16♠ 69♠	54 bytes of container in fragment (first one begins with two bytes of length of container starting from byte 3; last one is zero padded)	
70	Check Sum	

Fragment Control Byte

First Frame

This bit contains "1" if the packet carries the first frame of the container, otherwise zero.

Last Frame

This bit contains "1" if the packet carries the last frame of the container, otherwise zero.

Fragment number

Contains sequential number of the fragment, carried in the packet (starting from 1).

The byte structure of a container

The data structure to be fragmented and forwarded:

Forwarded Message Code							
Container (1) / Simple payload – 0	In case of container: sequential 7 bits ID of the container, otherwise - sequential 7 bits ID of the forwarded packet.						
Bit 7	Bit 6	Bit 6	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

1 2	Payload length (X)	2 bytes containing the length of the container starting from byte 3
3 3+X	The payload, X bytes (up to 512 bytes)	Received 3rd party or Garmin Packet
4+X	Unit's status + Current GSM Operator (1st nibble) (same as byte 16 of Msg type 0)	
5+X	Current GSM Operator (2nd and 3rd nibble) (same as byte 17 of Msg type 0)	
6+X	Current GSM Operator (4th and 5th nibble) (same as byte 25 of Msg type 0)	
7+X	Unit's mode of operation (same as byte 20 of Msg type 0)	

Copyright © 2013 by Pointer Telocation, Ltd.

8+X	Unit's I/O status 1st byte (same as byte 21 of Msg type 0)
9+X	Unit's I/O status 2nd byte (same as byte 22 of Msg type 0)
10+X	Unit's I/O status 3rd byte (same as byte 23 of Msg type 0)
11+X	Unit's I/O status 4th byte (same as byte 24 of Msg type 0)
12+X	Analog input 1 value (same as byte 26 of Msg type 0)
13+X	Analog input 2 value (same as byte 27 of Msg type 0)
14+X	Analog input 3 value (same as byte 28 of Msg type 0)
15+X	Analog input 4 value (same as byte 29 of Msg type 0)
16+X	Mileage counter (total 24 bits) (same as bytes 30-32 of Msg type 0)
17+X	
18+X	
19+X	Driver ID, PSP/SPC Specific Data, Accelerometer Status or SIM IMSI (same as bytes 33-38 of Msg type 0)
20+X	
21+X	
22+X	
23+X	
24+X	
25+X	Time of last GPS Fix (same as bytes 39-40 of Msg type 0)
26+X	
27+X	Location status (flags) (same as Sub-Type 4 of Msg type 9)
28+X	Mode 1 (from GPS)
29+X	Mode 2 (from GPS)
30+X	Number of satellites used (from GPS)
31+X	Longitude
32+X	
33+X	
34+X	
35+X	Latitude
36+X	
37+X	
38+X	
39+X	Altitude
40+X	
41+X	

Copyright © 2013 by Pointer Telocation, Ltd.

42+X	Ground speed
43+X	
44+X	Speed direction (true course)
45+X	
46+X	UTC time – seconds
47+X	UTC time – minutes
48+X	UTC time – hours
49+X	UTC date – day
50+X	UTC date - month
51+X	UTC date - year minus 2000 – 1 byte (e.g. value of 7 = year 2007)

2.6.2 *Create database structures to process the parsed data, create business reports and functions*

No further description.

2.6.3 *Implement the requests and setting of thresholds as type 5 messages*

Forward Data Command Definition (Message type 5)

Message Ingredients:

The forward data command has a varying length up to 217 bytes. It contains the following data (listed in the actual transmitted order):

Message header:

- System code – 4 bytes
- Message type – 1 byte
- Target Unit's ID – 4 bytes
- Command numerator – 1 byte
- Authentication code – 4 bytes
- Settings Byte – 1 byte
- Data length – 1 byte
- Data to Forward – variable up to 199 bytes
- Error detection code – 8-bit additive checksum (excluding system code)

Detailed Per-Field Specifications

Message header

Identical to Message header of Command type 0, except the Message type field is sent as 5 (five).

Copyright © 2013 by Pointer Telocation, Ltd.

Settings Byte

This byte is used for different system indications. For Squarell 8000 series integration, set this byte to '0'.

Data Length

This field should contain a number of bytes to forward (up to 199 bytes).

Data to Forward

This is the data that is forwarded to the terminal attached to the unit. This field must be an exact number of bytes long, as listed in the Data Length field.

Checksum

The checksum is a last byte of sum of all bytes in a message, excluding the 4 bytes of System Code and the Checksum itself.

Example:

The message:

*4D434750*0006000000081A0202120400000021006230000000 6B00E1000000000000000000000E5A100040206614EA303181A57034E12000000000000000000 1525071403D607*CS*

Calculation of the CS=>

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

=>CS=0x49

2.6.4 *Implement device health and configuration application for settings and thresholds*

No further description.

Note: There is an option to implement a server side application to upload Squarell firmware, as described in the "Upload protocol CORE devices XXXXXX.pdf".

Copyright © 2013 by Pointer Telocation, Ltd.

3 Squarell's 8000 Series RS232 Protocol

This section describes the RS232 messages and supported parameters in detail.

Parameters written in bold are stored data (in case of a power dip the values will NOT be lost).

Parameters written in Italic are NOT stored data (in case of a power dip the values will be lost).

IMPORTANT: Not all parameters from the list may be available. This availability of the data depends on the brand, model, year and configuration of the vehicle.

3.1 Messages Overview

These are the messages sent by the Squarell device:

$FMS1,<value1>,…,<value20><CR><LF> - Message will be frequented every 5sec.

$FMS2,<value1>,…,<value10><CR><LF> - On request

$FMS3,<value1>,…,<value17><CR><LF> - Send on event see the message description

$FMS4,<value1>,…,<value14><CR><LF> - Message will be frequented every 5 sec.

$FMS7,<value1>,…,<value17><CR><LF> - Future implementation.

$check,COM OK<CR><LF> - On request

$Version=<value1><CR><LF> - On request

$ExtendedVersion,<value1>,…,<value10><CR><LF> - On request

$Calibration,<value1><value2><CR><LF> - On request or confirming the speed and fuel calibration $

$SqTh1,<value1><value20><<CR><LF> - To confirm or to receive the Threshold Group 1 settings

$TrailerID, <value1><CR><LF> - Send on event or on Request

$TFUALL,<value1>,…,<value4><CR><LF> - On request

$FMSConfig,<value1>,…,<value20><CR><LF> - On request

$SqDiag1,<value1>,…,<value20><CR><LF> - On request

These are the messages that can be received by the Squarell device:

FMS2,1<CR><LF> - To request the $FMS2 data

FMS3,1<CR><LF> - To request the $FMS3 data

REset4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0<CR><LF> - To reset the $FMS4 data

SetKM,<value1><CR><LF> - To set the odometer

SqTFU,<value1><CR><LF> - To set the total fuel used

Copyright © 2013 by Pointer Telocation, Ltd.

SetTEH,<value1><CR><LF> - To set the engine hours

SqCalib,<value1>,<value2><CR><LF> - To set the speed and fuel calibration factors

SetTh1,<value1><value20><<CR><LF> - To set the Thresholds Group 1 remotely

SqTh1,1<CR><LF> - To request the Thresholds Group 1 settings

check,1<CR><LF> - To check the COM port communication

$Version,1<CR><LF> - To request the DCF Version

$ExtendedVersion,1<CR><LF> - To request the device info

Calibration,1<CR><LF> - To request the Speed and Fuel calibration factors

SqTID,1<CR><LF> - To request the Trailer ID

SqTFUConfig,<value1><CR><LF> - To configure the TFU output in $FMS1 message

SqTFUALL,1<CR><LF> - To request the different TFU values

SetFMS,<value1><CR><LF> - To configure the Squarell Messages

SqFMSConfig,1<CR><LF> - To request the settings of the FMS message timers

SqDiag1,1<CR><LF> - To request the Diagnostic message

These are the messages that can be used to obtain data from the Digital Tachograph:

$T, (To Tachograph)

$S, (From Tachograph)

3.1.1 *Detailed description of the $FMS1 message*

This message is sent every 5 seconds.

The format of the message is:

$FMS1,<value1>,…,<value20><CR><LF>

1.	**<Odometer> 1/200**	**total km vehicle distance**	**Count up**
2.	**<Total Fuel>* 1/2**	**total fuel used (litres)**	**Count up**
3.	**<Engine Hours> 1/20**	**hours**	**Count up**
4.	*<Actual Speed> 1/10*	*km /h*	*Real time*
5.	*<Actual Engine Speed*	*1/8 RPM*	*Real time*
6.	*<Actual Engine Torque>*	*1/1 & offset -125 % of maximum*	*Real time*
7.	*<Kickdown Switch>*	*0=no 1=yes*	*Real time*
8.	*<Accelerator Pedal Position>*	*4/10 %*	*Real time*
9.	*<Brake Switch>*	*0=no 1=yes*	*Real time*
10.	*<Clutch Switch>*	*0=no 1=yes*	*Real time*
11.	*<Cruise Active>*	*0=no 1=yes*	*Real time*
12.	*<PTO Active>*	*0=no >0=yes*	*Real time*
13.	*<Fuel Level>*	*4/10%*	*Real time*
14.	*<Engine Temperature>*	*1/1 & offset -40 Celsius*	*Real time*
15.	*<Turbo Pressure>*	*2/100 bar*	*Real time*
16.	*<Axle Weight 0>*	*½ kg*	*Real time*

Copyright © 2013 by Pointer Telocation, Ltd.

17.	*<Axle Weight 1>*	*½ kg*	*Real time*
18.	*<Axle Weight 2>*	*½ kg*	*Real time*
19.	*<Axle Weight 3>*	*½ kg*	*Real time*
20.	*<Service Distance>*	*5/1 km*	*Count down*

*** Total Fuel will present according to Squarell routines (Default setting and only available on 8000 Series).**

3.1.2 *Detailed description of the $FMS2 message*

This message is send after receiving the following request message: **FMS2,1<CR><LF>**.

The format of the message is:

$FMS2,<value1 >,…,<value10><CR><LF>

1.	*<Vehicle ID>*	*TEXT*	*17 characters*		*Static*
2.	*<Vehicle Registration Number> **	*TEXT*	*10 characters*		*Static*
3.	*<Year> **	*1/1*	*1985 to 2235*		*Real time*
4.	*<Month> **	*1/1*	*Month*		*Real time*
5.	*<Day> **	*¼*	*Day*		*Real time*
6.	*<Hours> **	*1/1*	*Hours*		*Real time*
7.	*<Minutes> **	*1/1*	*Minutes*		*Real time*
8.	*<Seconds> **	*¼*	*Seconds*		*Real time*
9.	*<Local Minute Offset> **		*1 min/bit & offset -125*	*-59 to +53*	*Real time*
10.	*<Local Hour Offset> **		*1hr/bit & offset -125*	*-23 to+23*	*Real time*

*Parameters are only available on the devices 6629-31, 6630-31 or 8000 Series.

3.1.3 *Detailed description of the $FMS3 message*

This message is sent 10 seconds (on the 8000 Serie 5 seconds) after an event on "Tacho Drive Recognise", "Tacho Work State Driver 1", "Tacho Work State Driver 2", "Tacho Overspeed", "Tacho Driver 1 Card Present", "Tacho Driver 2 Card Present", "Tacho Time State Driver 1", "Tacho Time State Driver 2" and "Tacho Direction" (an event is change of data). This message is also sent after receiving the following request message: **FMS3,1<CR><LF>**

The format of the message is:

$FMS3,<value1>,…,<value17><CR><LF>

1.	*<Tacho Drive Recognise>*	*0=no 1=yes*	*Real time*
2.	*<Tacho Work State Driver 1>*	*0=rest 1=available 2=work 3=drive*	*Real time*
3.	*<Tacho Work State Driver 2>*	*0=rest 1=available 2=work 3=drive*	*Real time*
4.	*<Tacho Overspeed >*	*0=no 1=yes*	*Real time*
5.	*<Tacho Driver 1 Card Present>*	*0=no 1=yes*	*Real time*
6.	*<Tacho Driver 2 Card Present>*	*0=no 1=yes*	*Real time*

Copyright © 2013 by Pointer Telocation, Ltd.

7.	*<Tacho Time State Driver 1>*	*0=normal 1=15m before 4,5 hrs 2=4,5 hrs 3=15m before 9 hrs, 4=9hrs, 5=15 min before 16 hrs, 6=16hrs,*	*Real time*
8.	*<Tacho Time State Driver 2>*	*0=normal 1=15m before 4,5 hrs 2=4,5 hrs 3=15m before 9 hrs, 4=9hrs, 5=15 min before 16 hrs, 6=16hrs,*	*Real time*
9.	*<Tacho Direction>*	*0=forward 1 =reverse*	*Real time*
10.	*<Tacho Performance>*	*0=normal 1=performance analysis*	*Real time*
11.	*<Tacho Handling Info>*	*0=no 1=yes*	*Real time*
12.	*<Tacho Event>*	*0=no event 1=event*	*Real time*
13.	*<Tacho Vehicle Speed>*	*1/256km/h*	*Real time*
14.	*<Driver 1 ID>*	** TEXT 17 characters*	*Static*
15.	*<Driver 2 ID>*	** TEXT 17 characters*	*Static*
16.	*<Odometer>*	** 1/200 total km vehicle distance*	*Count up*
17.	*<Total Fuel>*	** ½ total fuel used (litres)*	*Count up*

*Parameters are only available on the devices 6629-31, 6630-31 or 8000 Series.

3.1.4 *Detailed description of the $FMS4 message*

This message is sent every 5 seconds.

The format of the message is:

$FMS4,<value1>,…,<value14><CR><LF>

1.	*<Duration Driving>*	*seconds*	*Integer Count up*
2.	*<Duration Cruise Active>*	*seconds*	*Integer Count up*
3.	*<Idling Time>*	*seconds*	*Integer Count up*
4.	*<Fuel Used During Idle>*	*1/10 Litres*	*Non-integer (1 signification) Count up*
5.	*<Duration of PTO (stand still)>*	*seconds*	*Integer Count up*
6.	*<Fuel Used During PTO (stand still)>*	*1/10 Litres*	*Non-integer (1 signification) Count up*
7.	*<Number of Brake Applications>*	*times Integer*	*Count up*
8.	*<Num. of Brake Apps > THR Harsh Brk> * times*		*IntegerCount up*
9.	*<Num. of Clutch Presses>*	*times*	*Integer Count up*
10.	*<Idle Duration > THR Idle Dura> *times*		*IntegerCount up*
11	*<Duration Engine Torque > THR % Trq 4> * seconds*		*Integer Count up*
12.	*<Duration Speed > THR Spd 4> * seconds*		*Non-integer (1 signification) Count up*
13.	*<Duration RPM > THR RPM 4> * seconds*		*Integer Count up*
14.	*<Duration of Acceleration > THR Harsh Acc> * seconds*		*IntegerCount up*

* See the Thresholds section for more details.

3.1.5 *Detailed description of the REset4 message*

This message can be used to Set/Reset all the Parameters in the $FMS4 message.

The format of the message is:

REset4,<value1>,…,<value14><CR><LF>

Copyright © 2013 by Pointer Telocation, Ltd.

1.	*<Duration Driving>*	*seconds*	*Integer*	*Count up*
2.	*<Duration Cruise Active>*	*seconds*	*Integer*	*Count up*
3.	*<Idling Time>*	*seconds*	*Integer*	*Count up*
4.	*<Fuel Used During Idle>*	*1/10 Litres*	*Non-integer (1 signification)*	*Count up*
5.	*<Duration of PTO (stand still)>*	*seconds*	*Integer*	*Count up*
6.	*<Fuel Used During PTO (stand still)>*	*1/10 Litres*	*Non-integer (1 signification)*	*Count up*
7.	*<Number of Brake Applications>*	*times*	*Integer*	*Count up*
8.	*<Num. of Brake Apps > THR Harsh Brk>*	*times*	*Integer*	*Count up*
9.	*<Num. of Clutch Presses>*	*times*	*Integer*	*Count up*
10.	*<Idle Duration > THR Idle Dura>*	*times*	*Integer*	*Count up*
11	*<Duration Engine Torque > THR % Trq 4>*	*seconds*	*Integer*	*Count up*
12.	*<Duration Speed > THR Spd 4>*	*seconds*	*Non-integer (1 signification)*	*Count up*
13.	*<Duration RPM > THR RPM 4>*	*seconds*	*Integer*	*Count up*
14.	*<Duration of Acceleration > THR Harsh Acc>*	*seconds*	*Integer*	*Count up*

Example:

If you want to set all the parameters to 0 than send the following message to the Squarell device:

REset4,0,0,0,0,0,0,0,0,0,0,0,0,0,0<CR><LF>

3.1.6 *Detailed description of the SetKM message*

This message can be used to Set/Reset the Odometer.

The format of the message is:

SetKM,<value1><CR><LF>

1.	**<Odometer>**	**1/200**	**total km vehicle distance**	**Set value**

3.1.7 *Detailed description of the SqTFU message*

This message can be used to Set/Reset the Total Fuel.

The format of the message is:

SqTFU,<value1>,0<CR><LF>

1.	**<TFU in mL>**	**1/1000**	**total fuel used (litres)**	**Set value**

3.1.8 *Detailed description of the SetTEH message*

This message can be used to Set/Reset the Total Fuel.

The format of the message is:

SetTEH,<value1><CR><LF>

1.	**<Engine Hours>**	**1/20**	**hours**	**Set value**

Copyright © 2013 by Pointer Telocation, Ltd.

3.1.9 *Detailed description of the Check messages*

This message can be used to check if the COM port is OK and the RS232 communication is working without changing any settings inside the Squarell device.

The message is sent after receiving the following request message: **check,1<CR><LF>**.

The format of the message is:

$check,COM OK<CR><LF>

3.1.10 *Detailed description of the Version messages*

This message is sent after receiving the following request message: **$Version,1<CR><LF>**.

The format of the message is:

$Version=<value1><CR><LF>

1.	**<DCF Version>**	***TEXT***	***6* characters**	***Static***

3.1.11 *Detailed description of the Extended Version messages*

This message is send after receiving the following request message: **$ExtendedVersion,1<CR><LF>**.

The format of the message is:

$ExtendedVersion, <value1>,…,<value10><CR><LF>

1.	**<Device Code>**	***TEXT***	**7 characters**	***Static***
2.	**<Device Serial Number>**	**value**	**Integer**	***Static***
3.	**<DCF Name>**	***TEXT***	***XX* characters**	***Static***
4.	**<Device HW Version>**	**value**	**Integer**	***Static***
5.	**<Device SW Version>**	**value**	**Integer**	***Static***
6.	**<DDF Version>**	**value**	**Integer**	***Static***
7.	**<GDS Version>**	**value**	**Integer**	***Static***
8.	**<Firmware Version> ***	**value**	**Integer**	***Static***
9.	**<Future Use> ***			
10.	**<Future Use> ***			

* Only applicable to 8000 Series.

3.1.12 *Detailed description of the Calibration messages*

The Calibration Factors for Speed and Fuel can be modified or requested by RS232. See the *Calculating a Calibration Factor* section for information on how to calculate a factor.

To request the Calibration Factors the following message must be sent:

Calibration, 1<CR><LF>

Copyright © 2013 by Pointer Telocation, Ltd.

To set the Calibration factors the following message must be sent:

SqCalib,<value1>,<value2><CR><LF>

1.	**<Speed Calibration factor>**	**value**	**Integer**	**Set value**
2.	**<Fuel Calibration factor>**	**value**	**Integer**	**Set value**

3.1.13 *Detailed description of the Trailer ID messages*

This message is sent after receiving the following request message: **SqTID1, 1<CR><LF>** or when the brake pedal is depressed. The format of the message is:

$TrailerID, <value1><CR><LF>

1.	**<Trailer ID>**	**value**	**Integer**	**Static**

3.1.14 *Detailed description of the $FMS7 message (future implementation)*

This message is sent every 5 seconds.

The format of the message is:

$FMS7,<value1>,…,<value17><CR><LF>

1.	**<Total CO2 Emission >**	**26/20 Kg**	**Non-integer (1 signification) Count up**	**0 to 400000000**
2.	**<Future Use>**			
3.	**<Future Use>**			
4.	**<Future Use>**			
5.	**<Future Use>**			
6.	**<Future Use>**			
7.	**<Future Use>**			
8.	**<Future Use>**			
9.	**<Future Use>**			
10 .	**<Future Use>**			
11.	**<Future Use>**			
12.	**<Future Use>**			
13.	**<Future Use>**			
14.	**<Future Use>**			
15.	**<Future Use>**			
16.	**<Future Use>**			
17.	**<Future Use>**			

3.1.15 *Detailed description of the TFU Configuration messages*

This message is sent after receiving the following request message: **SqTFUALL,1<CR><LF>**

$TFUALL,<value1>…<value4><CR><LF>

1.	**< Squarell TFU 1 >**	**1/2**	**total fuel used (litres)**	**Count up**

Copyright © 2013 by Pointer Telocation, Ltd.

2.	**< Squarell TFU 2>**	**1/1000**	**total fuel used (litres)**	**Count up**
3.	**<Vehicle Dashboard TFU>**	**1/2**	**total fuel used (litres)**	**Count up**
4.	**<Vehicle dashboard TFU >**	**1/1000**	**total fuel used (litres)**	**Count up**

To configure the Total Fuel Used parameter in the $FMS1 message the following message must be sent:

SqTFUConfig,<value1><CR><LF>.

<value1> can be:

1= Squarell TFU with 0,5 litre resolution

2= Squarell TFU with 0,001 litres resolution

3= Dashboard TFU with 0,5 litres resolution

4= Dashboard TFU with 0,001 litres resolution

3.1.16 *Detailed description of the $FMS messages Configuration message*

This message is sent after receiving the following request message:
SqFMSConfig,1<CR><LF>

The fields in this message show the values of that particular $FMS message frequency time which will be sent out on every setup frequency.

$FMSConfig,<value1>,…,<value20><CR><LF>

1.	**< $FMS1 timer >**	**½**	**Seconds**	**Set value**
2.	**< $FMS2 timer>**	**½**	**Seconds**	**Set value**
3.	**< $FMS3 timer >**	**½**	**Seconds**	**Set value**
4.	**< $FMS4 timer >**	**½**	**Seconds**	**Set value**
5.	**<Future Use>**			
7.	**<Future Use>**			
8.	**<Future Use>**			
9.	**<Future Use>**			
10 .	**<Future Use>**			
11.	**<Future Use>**			
12.	**<Future Use>**			
13.	**<Future Use>**			
14.	**<Future Use>**			
15.	**<Future Use>**			
16.	**<Future Use>**			
17.	**<Future Use>**			
18.	**<Future Use>**			
19.	**<Future Use>**			
20.	**<Future Use>**			

To configure the frequency time of the $FMS messages the following message must be sent:

SetFMS,<value1>,…,<value20><CR><LF>

Copyright © 2013 by Pointer Telocation, Ltd.

1.	< $FMS1 timer >	½	Seconds	Set value
2.	< $FMS2 timer>	½	Seconds	Set value
3.	< $FMS3 timer >	½	Seconds	Set value
4.	< $FMS4 timer >	½	Seconds	Set value
5.	<Future Use>			
7.	<Future Use>			
8.	<Future Use>			
9.	<Future Use>			
10 .	<Future Use>			
11.	<Future Use>			
12.	<Future Use>			
13.	<Future Use>			
14.	<Future Use>			
15.	<Future Use>			
16.	<Future Use>			
17.	<Future Use>			
18.	<Future Use>			
19.	<Future Use>			
20.	<Future Use>			

The Default values are 10 which is 5s.

On both messages the Squarell device will reply with:

$Calibration,<value1>,<value2><CR><LF>

1.	<Speed Calibration factor>	value	Integer	Static
2.	<Fuel Calibration factor>	value	Integer	Static

3.1.17 *Detailed description of the $SqDiag message Diagnostic message*

This message is sent after receiving the following request message:
SqDiag1,1<CR><LF>

$SqDiag1,<value1>,…,<value20><CR><LF>

To reset the fields the following message must be sent:

ReSetDiag1,<value1>,…,<value20><CR><LF>

Example:

If you want to reset all the parameters send the following message to the Squarell device:

`ReSetDiag1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0<CR><LF>`

Copyright © 2013 by Pointer Telocation, Ltd.

4 Calculating a Calibration Factor

Some vehicles have incorrect Speed and Fuel data on their Vehicle CANbus. This will result in the wrong output from the Squarell device. To solve this problem Squarell uses Calibration Factors.

Default Speed and Fuel Calibration factors are 1000, which means that no calibration is done. The output can be calibrated by changing these factors, which results in the correct speed and fuel output.

NOTE: In order to calculate Calibration Factors, the real Speed and Fuel data must be known!

4.1 How to calculate a Calibration Factor

If the CANbus Speed of the vehicle is 1.5 times higher than the Real Speed there is a Calibration Factor needed. This can be calculated by:

Speed Calibration Factor = 1000 / (Vehicle Speed on the CANbus / Real Speed)

So if the Vehicle Speed on the CANbus is 1.5 times too high we can make the following calculation:

Speed Calibration Factor = 1000 / (1.5 / 1.0) = 667

You can also make the calculation with Speed values (km/h). If the CANbus Speed is 75 km/h and the Real Speed is 50 km/h, we can make the following calculation:

Speed Calibration Factor = 1000 / (75 / 50) = 667

Calculating the Fuel can be done the same way. The factor can be calculated by:

Fuel Calibration Factor = 1000 / (CANbus Fuel / Real Fuel)

If the CANbus Fuel Used is 20 litres but the Real Fuel Used is 34 litres, then we can make the following calculation:

Fuel Calibration Factor = 1000 / (20 / 34) = 1700

NOTE: The default Calibration Factors are 1000. The value 1000 will not have any calibration effect on the Speed and Fuel values (CANbus Speed = Real Speed and CANbus Fuel = Real Fuel).

4.1.1 *Check a Calibration Factor*

The Calibration factors can be checked on a Squarell Device by sending:

Calibration, 1<CR><LF>

The Squarell Device will reply with:

$Calibration,<Speed Calibration Factor>,<Fuel Calibration Factor><CR><LF>

Copyright © 2013 by Pointer Telocation, Ltd.

4.1.2 *Change a Calibration Factor*

The Calibration factors can be set on a Squarell Device by sending:

SqCalib,<Speed Calibration Factor>,<Fuel Calibration Factor><CR><LF>

The Squarell Device will reply with:

$Calibration,<Speed Calibration Factor>,<Fuel Calibration Factor><CR><LF>

Copyright © 2013 by Pointer Telocation, Ltd.

5 Thresholds

Thresholds are used to calculate some parameters from the $FMS4 message.

To change the Threshold settings, it is possible to modify the values in **Class 080 Constant** in the DCF file using iConfigure, or by sending a RS232 message to the device.

5.1 Changing the Threshold setting using iConfigure

IMPORTANT: Threshold values in the DCF are in engineering format. To change the thresholds in iConfigure, please calculate the real life thresholds into engineering values first. Examples are given below.

The default threshold settings for Trucks and Buses are (real life values in brackets):

1.	THR Harsh Brk	=	150	(100=1m/s2, 150=1.5m/s2, 200=2m/s2)
2.	THR Idle Dura	=	60	(60=1min)
3.	THR % Trq 4	=	90	(90=90%)
4.	THR Spd 4	=	23040	(23040=90km/h)
5.	THR RPM 4	=	13600	(13600=1700rpm)
6.	THR Harsh Acc	=	100	(100=1m/s2, 150=1.5m/s2, 200=2m/s2)

(Only for 8000 DCF from 130319)

The default threshold settings for Cars and Vans are (real life values in brackets):

1.	THR Harsh Brk	=	200	(100=1m/s2, 150=1.5m/s2, 200=2m/s2)
2.	THR Idle Dura	=	60	(60=1min)
3.	THR % Trq 4	=	90	(90=90%)
4.	THR Spd 4	=	25600	(25600=100km/h)
5.	THR RPM 4	=	22400	(22400=2800rpm)
6.	THR Harsh Acc	=	140	(100=1m/s2, 140=1.4m/s2, 200=2m/s2)

5.2 Changing the Threshold setting using a RS232 message

In the 8000 series thresholds can also be modified or requested by RS232 messages remotely. These values are real life values and **not engineering values**. To request the Threshold settings the following message must be sent:

SqTh1,1<CR><LF>

To set the Threshold settings the following message must be sent:

SetTh1,<value1>,…,<value20><CR><LF>

1.	**<THR Harsh Brake>**	**value**	**Integer Set value**
2.	**<THR Idle Duration>**	**value**	**Integer Set value**
3.	**<THR % Torque 4>**	**value**	**Integer Set value**
4.	**<THR Speed 4>**	**value**	**Integer Set value**
5.	**<THR RPM 4>**	**value**	**Integer Set value**
6.	**<THR Harsh Acceleration >**	**value**	**Integer Set value**

Copyright © 2013 by Pointer Telocation, Ltd.

7. **<Future Use>**
8. **<Future Use>**
9. **<Future Use>**
10. **<Future Use>**
11. **<Future Use>**
12. **<Future Use>**
13. **<Future Use>**
14. **<Future Use>**
15. **<Future Use>**
16. **<Future Use>**
17. **<Future Use>**
18. **<Future Use>**
19. **<Future Use>**
20. **<Future Use>**

On both messages the Squarell device will reply with:

$SqTh1,value1>,<value2><CR><LF>

1.	**<THR Harsh Brake>**	**value**	**Integer Set value**
2.	**<THR Idle Duration>**	**value**	**Integer Set value**
3.	**<THR % Torque 4>**	**value**	**Integer Set value**
4.	**<THR Speed 4>**	**value**	**Integer Set value**
5.	**<THR RPM 4>**	**value**	**Integer Set value**
6.	**<THR Harsh Acceleration>**	**value**	**Integer Set value**

7. **<Future Use>**
8. **<Future Use>**
9. **<Future Use>**
10. **<Future Use>**
11. **<Future Use>**
12. **<Future Use>**
13. **<Future Use>**
14. **<Future Use>**
15. **<Future Use>**
16. **<Future Use>**
17. **<Future Use>**
18. **<Future Use>**
19. **<Future Use>**
20. **<Future Use>**

Copyright © 2013 by Pointer Telocation, Ltd.

6 DCF Initialization Routine

6.1 Digital Tachograph Initialization

All Standard Squarell DCFs (for the 8000 Series) with a version number of 130109 or higher have a routine inside that automatically recognises the Tachograph brand. Every 5 seconds the DCF searches the connected Digital Tacho. When the DCF does not identify the tacho after 8 attempts it will read the available tachograph data from the CANbus. Please ensure that your Firmware version is Rev35 and higher.

Requesting the settings of the initialized Digital Tacho:

To request the Tacho settings, the following message must be sent:

TachoK,1<CR><LF>

The Squarell device will reply with:

TachoK,8,x,x,1<CR><LF>

Restarting the Digital Tacho initialisation or Forcing to a Digital Tacho brand:

By sending a RS232 message:

To restart the initialisation process automatically, send the following message:

SetTachoK,0,1,512<CR><LF>

To force the device to VDO settings, send the following message:

SetTachoK,8,2,512<CR><LF>

To force the device to Stoneridge settings, send the following message:

SetTachoK,8,3,512<CR><LF>

To force the device to Stoneridge settings, send the following message:

SetTachoK,8,4,512<CR><LF>

By performing a Manual reinitialize sequence in the truck:

During ignition on, press the brake pedal 12 times in 10 seconds.
You will then see that the green LED of the device will flash on and off.

Copyright © 2013 by Pointer Telocation, Ltd.