



**DIGITAL
COMMUNICATIONS
TECHNOLOGIES**

ANTARES SB

Manual

[For firmware version 5.30]

**GSM version:
GPS/GSM/GPRS/SMS
850/900E/1800/1900 Mhz**



Antares SBTM GSM User's Manual (FW 5.30)

Digital Communications Technologies

September 9, 2009

Copyright

This manual is copyrighted. All type of reproduction of its content must be authorized by *Digital Communications TechnologiesTM*.

Document information

Version: 1.00

Type: Release.

Date of release: 08-09-09

History:

| Version | Comments |
|-----------------------|---|
| 5.30-1.00 08-09-09 | First release for Antares FW 5.30. |
| | <p>Added info. on Garmin Devices Support.</p> <p>Added info. on ORBCOMM Satellite Modems Support.</p> <p>Added info. on OBD Support.</p> <p>Added info. on SMS Alias.</p> <p>Added info. on SMS Gateway.</p> <p>Added info. on Virtual Odometer.</p> <p>Added info. on Authentication Mechanism.</p> <p>Added info. on Event Machine.</p> <p>Added info. on Event Definition.</p> <p>Added info. on Event Message.</p> <p>Added info. on Reset message.</p> <p>Added info. on TAIP Console.</p> <p>Added info. on Regions Report.</p> <p>Added info. on Store & Forward Thresholds.</p> <p>Added info. on MDT Mode.</p> <p>Added info. on Driving Metrics.</p> <p>Added info. on Reporting Mode.</p> <p>Added info. on Diagnostic Message.</p> <p>Added info. on Silent Actions.</p> <p>Changed info. on GPS Module Specifications.</p> <p>Changed info. on Status.</p> <p>Changed info. on Extended-EV Message Formats.</p> <p>Changed info. on IMEI as ID.</p> <p>Changed info. on UDP Origin Port.</p> <p>Changed info. on UDP Server Port.</p> <p>Changed info. on Counters, Timers, Distancers.</p> <p>Changed info. on Registration Parameters.</p> <p>Changed info. on Firmware Upgrade (Over the air).</p> <p>Changed info. on Local Lock.</p> <p>Changed info. on Create Circular Region “Here”.</p> <p>Changed info. on Destinations’ set.</p> <p>Changed info. on TAIP Console Sniffer.</p> <p>Changed info. on Keep Alive.</p> <p>Changed info. on Voice Call End.</p> <p>Changed info. on Destinations’ Set.</p> <p>Changed info. on Altitude.</p> <p>Changed Limited Warranty info.</p> <p>Corrected example on Using Acceleration Signals.</p> |

Special Note

This manual applies to the GSM version of the *Antares SBTM*.

Firmware version

This manual applies to firmware version 5.30 of the GSM version of the *Antares SBTM*. You can consult the unit’s firmware version with the >QVR< TAIP command.

Regulatory Compliance

FCC

This product operates with Wavecoms Q24PL transmitter.

FCC Parts 22H and 24E are granted to the Wavecom Q24PL under FCC identifier 09EQ24PL001.

The antenna gain, including cable loss, must not exceed 3 dBi at 1900 MHz / 1.4 dBi at 850 MHz for mobile operating configurations and 7 dBi at 1900 MHz / 1.4 dBi at 850 MHz for fixed mounted operations, as defined in 2.1091 and 1.1307 of the rules for satisfying RF exposure compliance.

In addition, the antenna used for this device must be installed to provide a separation distance of at least 20 cm from all persons.

GARMIN AUTHORIZED PARTNER



Digital Communications Technologies LLC is a Garmin authorized partner. Founded in 1989, Garmin is a pioneer in Global Positioning System (GPS) devices and the worldwide leader in the design, manufacture and sale of GPS equipment. The company has built and sold millions of GPS products that serve the automotive, aviation, marine, consumer, wireless, OEM, and general recreation markets. For more information visit <http://www8.garmin.com/solutions/pnd/partners.jsp>

LIMITED WARRANTY

Digital Communications Technologies warrants the original purchaser that for a period of twelve (12) months from the date of purchase, the product shall be free of defects in materials and workmanship under normal use. During the warranty period, Digital Communications Technologies shall, at its option, repair or replace any defective product upon return of the product to its facilities, at no charge for labor and materials. Any replacement and/or repaired parts are warranted for the remainder of the original warranty or ninety (90) days, whichever is longer. The original owner must promptly notify Digital Communications Technologies in writing that there is defect in material or workmanship. Such written notice must be received in all events prior to expiration of the warranty period.

International Warranty

The warranty for international customers is the same as for any customer within the United States, with the exception that Digital Communications Technologies shall not be responsible for any customs fees, taxes, or VAT that may be due.

Warranty Procedure

To obtain service under this warranty, please return the item(s) in question to the point of purchase. All authorized distributors and dealers have a warranty program. Anyone returning goods to Digital Communications Technologies must first obtain an authorization number. Digital Communications Technologies will not accept any shipment whatsoever for which prior authorization has not been obtained.

Conditions to Void Warranty

This warranty applies only to defects in parts and workmanship relating to normal use. It does not cover:

- Damage incurred in shipping or handling
- Damage caused by disaster such as fire, flood, wind, earthquake or lightning
- Damage due to causes beyond the control of Digital Communications Technologies such as excessive voltage, mechanical shock or water damage
- Damage caused by unauthorized attachment, alterations, modifications or foreign objects
- Damage caused by peripherals unless such peripherals were supplied by Digital Communications Technologies
- Defects caused by failure to provide a suitable installation environment for the products
- Damage caused by use of the products for purposes other than those for which it was designed
- Damage from improper maintenance
- Damage arising out of any other abuse, mishandling or improper application of the products

Digital Communications Technologies liability for failure to repair the product under this warranty after a reasonable number of attempts will be limited to a replacement of the product, as the exclusive remedy for breach of warranty. Under no circumstances shall Digital Communications

Technologies be liable for any special, incidental, or consequential damages based upon breach of warranty, breach of contract, negligence, strict liability, or any other legal theory. Such damages include, but are not limited to, loss of profits, loss of the product or any associated equipment, cost of capital, cost of substitute or replacement equipment, facilities or services, down time, purchasers time, the claims of third parties, including customers, and injury to property.

Disclaimer of Warranties

This warranty contains the entire warranty and shall be in lieu of any and all other warranties, whether expressed or implied (including all implied warranties of merchantability or fitness for a particular purpose) And of all other obligations or liabilities on the part of Digital Communications Technologies. Digital Communications Technologies neither assumes nor authorizes any other person purporting to act on its behalf to modify or to change this warranty, nor to assume for it any other warranty or liability concerning this product. This disclaimer of warranties and limited warranty are governed by the laws of the State of Florida, USA.

WARNING

Digital Communications Technologies recommends that the entire system be completely tested on a regular basis. However, despite frequent testing, and due to, but not limited to, criminal tampering or electrical disruption, it is possible for this product to fail to perform as expected.

Out of Warranty Repairs

Digital Communications Technologies will at its option repair or replace out-of-warranty products which are returned to its factory according to the following conditions. Anyone returning goods to Digital Communications Technologies must first obtain an authorization number. Digital Communications Technologies will not accept any shipment whatsoever for which prior authorization has not been obtained. Products which Digital Communications Technologies determines to be repairable will be repaired and returned. A set fee which Digital Communications Technologies has predetermined and which may be revised from time to time, will be charged for each unit repaired. Products which Digital Communications Technologies determines not to be repairable will be replaced by the nearest equivalent product available at that time. The current market price of the replacement product will be charged for each replacement unit.

Contents

| | | |
|----------|---|-----------|
| 0.1 | Scope | 19 |
| 0.2 | Organization | 19 |
| 0.3 | Technical Assistance | 20 |
| 1 | About the <i>Antares SBTM</i> | 21 |
| 1.1 | Features | 21 |
| 1.1.1 | Analog to Digital Converter | 21 |
| 1.1.2 | Discrete Inputs and Outputs | 21 |
| 1.1.3 | Optional Back-up Battery | 22 |
| 1.1.4 | Voice support | 22 |
| 1.1.5 | SMS support | 22 |
| 1.1.6 | TCP and UDP support | 22 |
| 1.1.7 | Over the air control/consult | 22 |
| 1.1.8 | Over the air upgrade | 22 |
| 1.1.9 | Versatile RS-232 communication | 23 |
| 1.1.10 | Communication buffer | 23 |
| 1.1.11 | Automatic outputs' control | 23 |
| 1.1.12 | DNS lookup | 23 |
| 1.1.13 | Turn-by-turn report | 23 |
| 1.1.14 | Driving metrics | 23 |
| 1.1.15 | GPS Back Log and Acceleration | 24 |
| 1.1.16 | IMEI as ID | 24 |
| 1.1.17 | Cell ID reporting | 24 |
| 1.1.18 | Molex-type connectors | 24 |
| 1.1.19 | SMA Reverse polarity GPS antenna connector | 24 |
| 1.1.20 | Event Reporting | 24 |

| | | |
|----------|---|-----------|
| 1.2 | Contents of package | 26 |
| 1.3 | Front side description | 27 |
| 1.3.1 | RS-232 port | 27 |
| 1.3.2 | SIM card slot | 28 |
| 1.3.3 | LEDs | 28 |
| 1.3.4 | AUDIO jack | 29 |
| 1.4 | Back side description | 30 |
| 1.4.1 | GSM Antenna connector | 30 |
| 1.4.2 | GPS Antenna connector | 30 |
| 1.4.3 | I/O molex-type connector | 30 |
| 1.4.4 | Power/ignition molex-type connector | 31 |
| 1.5 | Specifications | 33 |
| 1.5.1 | Dimensions | 33 |
| 1.5.2 | Environment | 33 |
| 1.5.3 | Power | 33 |
| 1.5.4 | Inputs/Ignition | 34 |
| 1.5.5 | Outputs | 34 |
| 1.5.6 | Analog To Digital Converter | 34 |
| 1.5.7 | Audio | 34 |
| 1.5.8 | GSM/GPRS module | 35 |
| 1.5.9 | GPS module | 36 |
| 1.5.10 | GSM antenna connector | 37 |
| 1.5.11 | GPS antenna connector | 38 |
| 2 | Installation | 39 |
| 2.1 | Power Supply | 39 |
| 2.1.1 | Vehicles with a main power switch | 39 |
| 2.2 | Inputs detection | 43 |
| 2.3 | Ignition detection | 43 |
| 2.4 | Outputs | 44 |
| 2.5 | Connection diagrams | 46 |

| | | |
|----------|---|-----------|
| 3 | Operation | 48 |
| 3.1 | Serial Port | 49 |
| 3.2 | LEDs | 49 |
| 3.2.1 | Power (Red) | 50 |
| 3.2.2 | Signal (Orange) | 50 |
| 3.2.3 | Fix (Yellow) | 51 |
| 3.2.4 | On line (Green) | 51 |
| 3.3 | Inputs/Ignition | 52 |
| 3.4 | Outputs | 53 |
| 3.5 | Analog to Digital Converter | 54 |
| 3.6 | Back-up Battery | 55 |
| 3.7 | Sleep Mode | 57 |
| 3.8 | Over The Air | 57 |
| 3.8.1 | Via IP hosts (GPRS) | 57 |
| 3.8.2 | Via SMS (GSM) | 60 |
| 3.8.3 | Voice (GSM) | 61 |
| 3.9 | TAIP console | 61 |
| 3.9.1 | TAIP Message Format | 61 |
| 3.9.2 | Reporting messages | 64 |
| 3.9.3 | Interacting | 64 |
| 3.10 | Remote host software | 64 |
| 3.10.1 | Working with TCP | 65 |
| 3.10.2 | Working with UDP | 66 |
| 3.10.3 | Working with SMS | 67 |
| 3.11 | Reports' messages | 68 |
| 3.11.1 | Events' Reporting Messages | 68 |
| 3.11.2 | Responses to TAIP Commands Messages | 69 |
| 3.12 | Reports' buffer | 70 |
| 3.13 | GPS Back Log | 71 |
| 3.14 | Virtual Odometer | 71 |
| 3.15 | Authentication Mechanism | 71 |

| | | |
|----------|--|-----------|
| 3.16 | SMS Alias | 72 |
| 3.17 | SMS Messages Gateway | 72 |
| 3.18 | Garmin Devices Support | 72 |
| 3.18.1 | Setting up <i>Antares SBTM</i> | 72 |
| 3.18.2 | Data Flow Example | 74 |
| 3.18.3 | Setting Up The Server | 79 |
| 3.19 | ORBCOMM Satellite Modems Support | 79 |
| 3.19.1 | Setting up <i>Antares SBTM</i> for ORBCOMM Satellite Modems | 79 |
| 3.19.2 | Operation | 80 |
| 3.19.3 | Example | 80 |
| 3.20 | OBD Support | 82 |
| 3.21 | Firmware Upgrade | 83 |
| 3.21.1 | Over The Air | 83 |
| 3.21.2 | Upgrading locally | 85 |
| 3.22 | <i>TAIP DownloaderTM</i> Tool (Write/Read scripts) | 85 |
| 3.22.1 | Communicating locally with the <i>Antares SBTM</i> | 86 |
| 3.22.2 | STEP 1. Selecting a COM port | 86 |
| 3.22.3 | STEP 2. Test Communication | 86 |
| 3.22.4 | Write a Configuration Script | 87 |
| 3.22.5 | Read a Configuration Script | 88 |
| 3.22.6 | Over The Air | 88 |
| 4 | Configuration | 89 |
| 4.1 | *Unit's ID | 90 |
| 4.2 | *Enabling the unit on GSM and GPRS | 90 |
| 4.2.1 | SIM Card's PIN for GSM registration | 91 |
| 4.2.2 | Access Point Name (APN) for GPRS set up | 92 |
| 4.3 | *Destinations (DPs and DAs) | 93 |
| 4.3.1 | Destination Points (DPs) | 93 |
| 4.3.2 | Destination Addresses (DAs) | 94 |

| | | |
|--------|---|-----|
| 4.4 | Reporting | 95 |
| 4.5 | *Event Machine | 96 |
| 4.5.1 | Triggers | 96 |
| 4.5.2 | Actions | 98 |
| 4.5.3 | Events | 101 |
| 4.5.4 | Signals | 102 |
| 4.5.5 | Examples | 107 |
| 4.6 | Using <i>Polygonal Regions</i> | 107 |
| 4.7 | Using <i>Circular Regions</i> (geo-fences) | 108 |
| 4.8 | Using Region ID Reports | 109 |
| 4.9 | Setting <i>Speed Limits</i> | 109 |
| 4.10 | The <i>Time And Distance</i> criteria | 110 |
| 4.11 | Using <i>Time Windows</i> | 111 |
| 4.12 | Using <i>Counters</i> | 111 |
| 4.13 | Manipulating <i>signals</i> | 116 |
| 4.14 | User <i>signals</i> | 116 |
| 4.15 | Using <i>Heading Deltas</i> (turn-by-turn report) | 116 |
| 4.16 | Driving Metrics (Acceleration, Max. Speed, etc) . . . | 117 |
| 4.17 | Using Acceleration <i>signals</i> | 119 |
| 4.18 | Voice calls | 120 |
| 4.19 | <i>Battery</i> monitoring | 120 |
| 4.20 | Serial port devices | 121 |
| 4.21 | <i>Analog to Digital Converter</i> monitoring | 123 |
| 4.22 | Using a TCP/UDP keep-alive | 123 |
| 4.23 | IMEI as ID | 123 |
| 4.24 | Cell ID reporting | 124 |
| 4.25 | Sleep mode | 125 |
| 4.26 | Restoring the unit | 125 |
| 4.27 | Resetting the unit | 125 |
| 4.28 | Using Scripts | 126 |
| 4.28.1 | Creating an script from scratch | 127 |

| | | |
|----------|--|------------|
| 4.28.2 | Reading Scripts | 127 |
| 4.28.3 | Writing Scripts | 128 |
| 4.28.4 | Scripts Over The Air | 128 |
| 5 | Scenarios and examples | 130 |
| 5.1 | Getting Started | 130 |
| 5.1.1 | Setting the unit's ID | 130 |
| 5.1.2 | Setting the APN and PIN | 130 |
| 5.1.3 | Creating a Destination Point (DP) | 131 |
| 5.1.4 | Creating a Destination Address (DA) | 132 |
| 5.1.5 | Creating a time-period criterion | 132 |
| 5.1.6 | Tiding a signal to an event | 133 |
| 5.1.7 | Checking the host software/server | 133 |
| 5.1.8 | Adding an Input report | 134 |
| 5.1.9 | Script | 134 |
| 5.2 | Adding SMS reporting | 135 |
| 5.2.1 | Create the SMS Destination Point | 135 |
| 5.2.2 | Create a new <i>Destination Address</i> | 135 |
| 5.2.3 | Change the Input report event definition | 136 |
| 5.2.4 | Create a SMS custom message | 136 |
| 5.2.5 | Check the reported message | 136 |
| 5.2.6 | Script | 137 |
| 5.3 | Adding SMS interaction | 138 |
| 5.3.1 | Query the unit with a SMS | 138 |
| 5.3.2 | Set an output with a SMS | 138 |
| 5.4 | Adding voice interaction | 139 |
| 5.4.1 | Make the unit accept a phone call | 139 |
| 5.4.2 | Have the unit initiate a voice call | 139 |
| 5.5 | Ignition detection | 140 |
| 5.5.1 | Script | 140 |
| 5.6 | Speed violation (with warning) report | 141 |

| | | |
|--------|--|-----|
| 5.6.1 | Setting the speed limit | 142 |
| 5.6.2 | Start a counter | 142 |
| 5.6.3 | Creating the violation report | 142 |
| 5.6.4 | Something is missing... | 142 |
| 5.6.5 | Driving the LED | 142 |
| 5.6.6 | Script | 143 |
| 5.7 | START/STOP monitoring | 144 |
| 5.7.1 | Setting a low speed limit | 145 |
| 5.7.2 | Start a counter | 145 |
| 5.7.3 | Create the STOP report | 145 |
| 5.7.4 | Create the START report | 146 |
| 5.7.5 | Something's missing... | 146 |
| 5.7.6 | Script | 146 |
| 5.8 | Safe engine turn off | 148 |
| 5.8.1 | Create the speed limit | 149 |
| 5.8.2 | Creating a timer | 149 |
| 5.8.3 | Cutting the ignition | 149 |
| 5.8.4 | Stopping the counter | 149 |
| 5.8.5 | Restore the user signal | 149 |
| 5.8.6 | Script | 150 |
| 5.8.7 | Operation | 152 |
| 5.9 | Improving the periodic report | 153 |
| 5.9.1 | Script | 154 |
| 5.10 | Reconnection event for TCP | 157 |
| 5.10.1 | Script | 157 |
| 5.11 | Main-power-loss alarm | 160 |
| 5.11.1 | Script | 160 |
| 5.12 | Using the sleep mode | 160 |
| 5.13 | Configuring/reading a distance counter | 161 |
| 5.14 | Generating an extended-EV report | 161 |

| | | |
|----------|---|------------|
| 6 | Unit's TAIP reference | 163 |
| 6.1 | (AL) Altitude | 164 |
| 6.2 | (CP) Compact Position | 165 |
| 6.3 | (DA) Destination Address | 166 |
| 6.3.1 | Examples | 167 |
| 6.4 | (DP) Destination Point | 168 |
| 6.5 | (ED) Event Definition | 170 |
| 6.5.1 | Examples | 172 |
| 6.6 | (ER) Error Report | 174 |
| 6.6.1 | Example | 174 |
| 6.7 | (ET) Event Report, time only message | 175 |
| 6.8 | (EV) Event Message | 176 |
| 6.9 | (GC) Counters, Timers, Distancers | 179 |
| 6.9.1 | Counters' commands | 180 |
| 6.9.2 | Examples | 180 |
| 6.10 | (GF) GPIOs' function (I/O) | 182 |
| 6.11 | (GS) Speed Limit | 183 |
| 6.12 | (GT) Time Window | 184 |
| 6.13 | (GR) Regions | 185 |
| 6.13.1 | Special cases | 186 |
| 6.13.2 | Regions' creation examples | 186 |
| 6.14 | (ID) Identification | 191 |
| 6.15 | (MS) Memory Session | 192 |
| 6.16 | (MT) MDT Mode | 193 |
| 6.17 | (PV) Position-velocity | 194 |
| 6.18 | (RF) Radio Frequency module configuration | 195 |
| 6.19 | (RM) Reporting Mode | 196 |
| 6.20 | (RP) Registration Parameters (Cellular Network) | 197 |
| 6.21 | (RT) Reset message | 199 |
| 6.22 | (SS) Signal Status | 200 |
| 6.22.1 | Examples | 200 |

| | |
|---|-----|
| 6.23 (ST) Status | 202 |
| 6.24 (TM) Time and Date | 203 |
| 6.25 (TD) Time and Distance signals configuration | 204 |
| 6.26 (TX) Text Message | 206 |
| 6.26.1 Escape sequences | 206 |
| 6.26.2 Garmin Mode Messages | 206 |
| 6.27 (VR) Version number | 211 |
| 6.28 (XAAC) Analog to Digital converter | 212 |
| 6.29 (XAAU) Challenge Text | 213 |
| 6.30 (XABS) Battery Status | 214 |
| 6.31 (XACE) Cell Environment | 215 |
| 6.32 (XACR) Counter Report | 216 |
| 6.32.1 Reported Message | 216 |
| 6.33 (XACT) Communication Test | 217 |
| 6.34 (XADM) Diagnostic Message | 218 |
| 6.35 (XADP) Destination Points | 221 |
| 6.35.1 IP-type destinations | 221 |
| 6.35.2 Telephone destinations | 221 |
| 6.36 (XAEF) Extended-EV message Formats | 223 |
| 6.37 (XAFU) Firmware Upgrade (Over the air) | 225 |
| 6.38 (XAGA) ADC levels | 226 |
| 6.39 (XAGB) Back-up Battery levels | 227 |
| 6.40 (XAGF) Store & Forward Thresholds | 228 |
| 6.41 (XAGH) Heading deltas | 229 |
| 6.42 (XAGM) Garmin Mode | 230 |
| 6.42.1 (XAGMI) Consult Garmin Device General Infor- mation | 231 |
| 6.42.2 (XAGMKI) Garmin Mode Driver ID | 231 |
| 6.42.3 (XAGMKSA) Garmin Mode Add Driver Status | 232 |
| 6.42.4 (XAGMKSD) Garmin Mode Delete Driver Status | 233 |
| 6.42.5 (XAGMKS) Garmin Mode Change Driver Status | 233 |

| | |
|---|-----|
| 6.42.6 (XAGMR) Garmin Mode Add or Delete Canned Replies | 234 |
| 6.42.7 (XAGMRS) Garmin Mode Canned Reply Text Message | 234 |
| 6.42.8 (XAGMS) Garmin Mode Stop Message | 235 |
| 6.42.9 (XAGMCS) Garmin Mode Change Stop Message Status | 237 |
| 6.42.10 (XAGMTS) Garmin Mode Text Send | 238 |
| 6.42.11 (XAGMT) Garmin Mode Message Status | 239 |
| 6.42.12 (XAGMTA) Garmin Mode Set Canned Message | 240 |
| 6.42.13 (XAGMTD) Garmin Mode Delete Canned Message | 240 |
| 6.42.14 (XAGMX) Delete Fleet Management Protocol Related Data | 241 |
| 6.43 (XAGN) Acceleration Limits | 242 |
| 6.44 (XAGP) GPRS Pause | 243 |
| 6.45 (XAGR) Circular Regions | 244 |
| 6.46 (XAID) IMEI as ID | 245 |
| 6.47 (XAIM) IMEI consult | 246 |
| 6.48 (XAIO) Input, Outputs consult | 247 |
| 6.49 (XAIP) IP address | 248 |
| 6.50 (XAIR) Create Circular Region “here” | 249 |
| 6.51 (XAIT) Driving Metrics | 250 |
| 6.52 (XAKA) Keep Alive | 251 |
| 6.53 (XAKL) GPS Back Log | 252 |
| 6.54 (XALL) Local Lock | 253 |
| 6.55 (XAMD) MD5 Check | 254 |
| 6.56 (XANB) Network Band mode | 255 |
| 6.57 (XANS) Network Status (GPRS) | 256 |
| 6.58 (XAOE) Engine’s RPM thresholds. | 257 |
| 6.59 (XAOF) Fuel Level percentage values. | 258 |
| 6.60 (XAOG) Remaining Fuel Gallons thresholds. | 259 |
| 6.61 (XAOR) Fuel Rate thresholds | 260 |

| | |
|--|------------|
| 6.62 (XAOS) OBD Status Consult | 261 |
| 6.63 (XAOT) Throttle Position thresholds. | 263 |
| 6.64 (XAPM) Power Management | 264 |
| 6.64.1 Examples | 266 |
| 6.65 (XAPW) Set Password | 267 |
| 6.66 (XARD) Reset diagnostics | 268 |
| 6.67 (XARE) Regions Report | 269 |
| 6.68 (XASD) Destinations' Set | 270 |
| 6.69 (XASF) Store & Forward Buffer | 271 |
| 6.70 (XASG) SMS Messages Gateway | 272 |
| 6.71 (XASI) IMSI Consult | 273 |
| 6.72 (XATA) SMS Alias | 274 |
| 6.73 (XATD) Current Destination Point | 275 |
| 6.74 (XATM) User-defined Text Messages | 276 |
| 6.75 (XATS) TAIP Console Sniffer | 277 |
| 6.75.1 Example | 277 |
| 6.76 (XAUN) UDP Network | 278 |
| 6.77 (XAUN) UDP Origin Port | 279 |
| 6.78 (XAUP) UDP Server Port | 280 |
| 6.79 (XAVC) Voice Call Start | 281 |
| 6.80 (XAVE) Voice Call End | 282 |
| 6.81 (XAVI) Voice Call Identification switch | 283 |
| 6.82 (XAVM) Microphone gain | 284 |
| 6.83 (XAVO) Virtual Odometer | 285 |
| 6.84 (XAVS) Speaker volume | 286 |
| 6.85 Errors list | 287 |
| | |
| 7 Appendix A - Quick Start Guide | 291 |
| | |
| 8 Appendix B - Getting Started Script | 294 |
| | |
| 9 Appendix C - Signals' Table | 295 |

| | |
|--|------------|
| 10 Appendix D - Quick TAIP reference | 299 |
| 10.1 Setting the <i>Antares SBTM</i> ID | 299 |
| 10.2 Setting the APN | 299 |
| 10.3 Configuring the SIM card PIN | 299 |
| 10.4 Restarting the unit | 299 |
| 10.5 Restoring to factory defaults | 299 |
| 10.6 Reseting the GPRS connection | 300 |
| 10.7 Configuring a host address/type | 300 |
| 10.8 Configuring a telephone number for SMS and Voice interaction | 300 |
| 10.9 Querying hosts/ports and telephones | 300 |
| 10.10 Grouping AVL servers on <i>DAs</i> | 301 |
| 10.11 Defining a periodic timer | 301 |
| 10.11.1 Using a time counter | 301 |
| 10.11.2 Using a Time And Distance counter | 301 |
| 10.12 Creating an event | 301 |
| 10.13 Creating a turn-by-turn (heading change) report . . . | 301 |
| 10.14 Creating a kilometer counter | 302 |
| 10.15 Setting an output | 302 |
| 10.16 Querying the state of an input | 302 |
| 10.17 Querying the state of the vehicle-ignition input | 302 |
| 10.18 Querying the Analog to digital converter | 302 |
| 10.19 Querying the Internal back-up battery level | 303 |
| 10.20 Driving the unit to sleep power mode | 303 |
| 10.21 Querying the firmware version | 303 |
| 10.22 Activating PAD mode on serial port | 303 |

Preface

This document is the *Antares SBTM* User's Guide. On this document you will find information on what is the *Antares SBTM*, its features, specifications, installation instructions and explanation on the unit's configuration and operation.

This document is available at:
<http://www.digitalcomtech.com>

Refer to this site or to your *Digital Communications TechnologiesTM* contact for the latest version of this document.

0.1 Scope

Most of the technical information related to the *Antares SBTM* device is expected to be written on this manual. However, there are some external documents called *Application Notes* which contains some specific development, that falls beyond the scope of this document.

This manual is intended to be used by anyone interacting with the unit and having some basic technical knowledge.

After reading this document the reader will be capable to install, configure and operate the unit on the day-to-day vehicle tracking job.

0.2 Organization

This document is organized in the following way:

- The *About* chapter gives a functional and physical description of the unit.
- The *Installation* chapter has guides and recommendations on the physical and electrical conditions for the installation of the unit.
- The *Operation* chapter gives information on how to interact with the unit.
- The *Configuration* chapter instructs on how to configure the unit.

- The *TAIP reference* chapter is a compendium of all the configuration and query commands, therefore it is the big complement of the *Operation* and *Configuration* chapters.
- The *Quick Start Guide* is a very condensed summary to get you started with the unit.

0.3 Technical Assistance

You can contact *Digital Communications Technologies*TM for technical support at:

`support@digitalcomtech.com`

Or by calling
1 305 7183336
9AM to 5PM Eastern US time.

1 About the *Antares SBTM*

The *Antares SBTM* is a vehicle tracking and controlling device designed to interact remotely with Automated Vehicle Location (AVL) systems or end-users by using the GSM/GPRS cellular network as communication media. The Antares unit is installed on a vehicle whose geographical position and/or state is desired to be remotely monitored/controlled.

The geographical position is taken from the unit's built-in GPS receiver which gives information such as position, velocity, heading, time-date, acceleration, altitude. The vehicle's state may be monitored and/or controlled by using the unit's discrete inputs-outputs, analog-to-digital converter, audio support and its RS-232 serial port. The last one useful to communicate with expanding accessories such as PDAs or MDTs¹.

1.1 Features

A list with the unit's features is presented next. A brief descriptions is given, for detailed information see the given sections/chapters.

1.1.1 Analog to Digital Converter

An input voltage ranging between 0 and 32V may be measured with the ADC. For information on the ADC refer to the *Analog to Digital Converter* section on the *Operation* chapter.

1.1.2 Discrete Inputs and Outputs

The unit has 4 discrete inputs, 4 discrete outputs and an ignition sensor.

Electrical information is found on the *About* and *Operation* chapters.

¹ Mobile Data Terminal.

1.1.3 Optional Back-up Battery

The *Antares SBTM* may include² a built-in back-up battery to be used when the vehicle's battery is unavailable. Refer to the *Operation* chapter for more information.

1.1.4 Voice support

An audio jack for non-balanced hands-free audio systems allows the unit to initiate and receive phone calls (Hands-free audio system is not included). Refer to the *Operation* and *Configuration* chapters.

1.1.5 SMS support

When the unit is registered on the GSM network it can send and receive SMSs. This feature is used to send user-defined event's text to phone numbers, TAIP reports to SMPP servers and to receive commands or queries to interact with the unit over-the-air. See the *Operation* and *Configuration* chapters.

1.1.6 TCP and UDP support

The unit may send its reports via GPRS to IP hosts using TCP and/or UDP transport protocols. As an improvement from previous versions, all IP-type *Destinations* can be used either on TCP or UDP and the unit may work with *Destinations* on TCP and with *Destinations* on UDP at the same time. This means that a global parameter (*XASP*) defining the transport protocol for all DPs no longer exists.

Note:

| |
|---|
| The DP and XADP TAIP messages have been modified to support this new feature. |
|---|

1.1.7 Over the air control/consult

The unit can be controlled/consulted remotely via GPRS (TCP or UDP) and/or via GSM by means of SMS messages.

1.1.8 Over the air upgrade

The unit's firmware may be upgraded³ via GPRS communication with a single instruction.

² Ask for built-in battery when buying the unit.

³ Not all units have this feature enabled. TAIP error 69 or 90 is returned when using the firmware upgrade command (*XAFU*)

1.1.9 Versatile RS-232 communication

The unit's serial port can be used to configure/control the unit and it may also be used to transfer any byte-like messages to and from remote *Destinations*. One application for this is attaching an MDT⁴ device. Other example includes attaching a satellital modem to be used when no GSM/GPRS signal is detected by the *Antares SBTM*.

1.1.10 Communication buffer

Antares SBTM will start saving event's reports and incoming serial port data whenever one or more *Destinations* are unreachable.

1.1.11 Automatic outputs' control

The outputs can be driven by commands or they may be driven automatically by the unit whenever a pre-configured situation occurs.

1.1.12 DNS lookup

IP-type *Destinations* can be defined with a numeric IP address or with a host name. *Antares SBTM* will use the carrier's DNS servers to resolve names. This feature is very useful when the IP-host(s) resides on an IP-changing environment.

1.1.13 Turn-by-turn report

By tracking the vehicle's heading change, a turn-by-turn report can be achieved. This leads to detailed tracking of a vehicle's route and also to a reduction of unnecessary reports on long straight roads and highways.

1.1.14 Driving metrics

The vehicle's instant acceleration can be obtained at any time, and also, the maximum acceleration and maximum speed values with their respective GPS location can be saved and reported, so that 'good' and 'bad' drivers can be more easily detected. The maximum negative acceleration value gives information on the maximum break-force applied, the maximum positive acceleration gives information on gas pedal usage and the maximum speed aids in controlling safety and controlling vehicle's stress.

⁴ Mobile Data Terminal: Vehicle's device that ables an interaction between a vehicle's crew and an AVL facility.

1.1.15 GPS Back Log and Acceleration

Positive and negative accelerations can be monitored to generate reports on large gas pedal usage and breaking/crashing conditions. Also a GPS Back Log that stores all data received from the internal GPS module at a 1-second rate can be retrieved at any time, for example when a large negative acceleration (possible crash) is detected. With this log, the last minute of the vehicle's location/speed can be examined second by second.

1.1.16 IMEI as ID

This feature allows the unit to tag every reported message with the unit's IMEI. This eases the management of unit's as the ID number does not require to be programmed and it is a unique number that can't be reused or shared with another unit. Also, this number can not be deleted or changed.

1.1.17 Cell ID reporting

Antares SBTM can add the Cellular Network Cell ID information on every reported message. This enables a Tracking System to locate the unit when GPS is not available. *Antares SBTM* will report the Cell ID, LAC, MCC, MNC and RSSI of the cell it is registered with. This information can be used by systems that know the location of Cells to approximate a location of an unit with no GPS.

1.1.18 Molex-type connectors

Inputs, outputs, ADC input voltage, ignition sense and power are all provided on molex-type male connectors which allow for molex-type female receptacles. This quality industry-proved type of connectors is very suitable for vehicles' environments.

1.1.19 SMA Reverse polarity GPS antenna connector

The antennas can not be erroneously interchanged because of the reverse-polarity condition of the GPS antenna connector.

1.1.20 Event Reporting

Antares SBTM has the ability to interpret complex user-defined reporting criteria to track normal, as well as exceptional situations. This is called *Event Report* and it is mainly done via the *Event Machine* included in the unit's firmware.

This allows to create scenarios that include the boolean combination of the following variables:

1.1. FEATURES

- 30 Polygon-defined geographical regions (50 points each).
- 100 circular geographical regions.
- 10 Speed limits.
- 5 Positive/negative acceleration limits.
- 10 Time windows (dates' intervals).
- 4 Discrete inputs
- 4 Discrete outputs
- 20 Counters for traveled distance, time and event counting.
- 5 Heading change deltas (turn by turn report).
- 5 Analog to Digital Converter thresholds.
- 5 Back-up battery level thresholds.
- 10 User signals to create complex reports.
- Fixed signals:
 - Vehicle Ignition.
 - Main power detection.
 - 12volts/24volts detector for main power.
 - GPS Fix state.
 - GSM roaming state.
 - GSM registration state.
 - GPRS registration state.
 - GPRS attach state.
 - GPS Antenna short circuit state.
 - TCP connections' state indicators.
 - Software reset indicator.
 - Voice call state indicators.
 - Woke Up Signal

1.2 Contents of package

Inside the *Antares SBTM* box you will find the following content:



- An *Antares SBTM*.
- A GSM Quad-Band antenna ready to work with any GSM carrier regardless of its operation frequency.
- An active GPS antenna with magnetic support and reverse-polarity connector.
- The I/O harness: 10 color-coded cables 1 meter (3.28 feet) long attached to a female molex-type receptacle on the unit's side and open ends on the other.
- The Power harness: 3 color-coded cables 1 meter (3.28 feet) long attached to a female molex-type receptacle on the unit's side and open ends on the other.

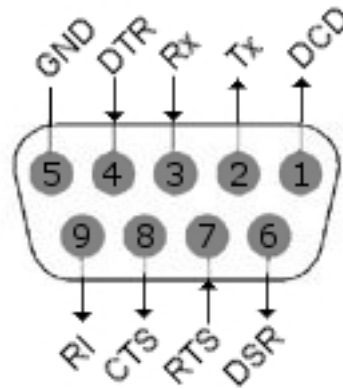
1.3 Front side description



1.3.1 RS-232 port

DB9 female connector with all of the RS-232 signals available for serial communication.

The DB9 signals' pin-out is:



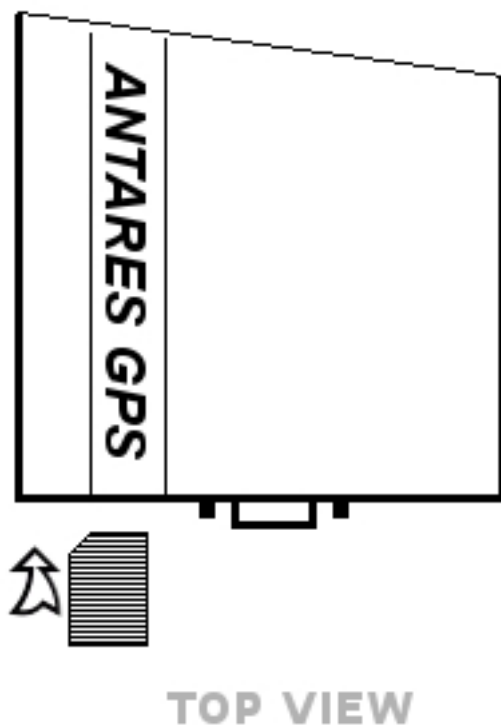
Use this port to configure or query the unit and to connect accessories like PDA-like devices or MDTs.

The *Antares SBTM*'s works as an RS-232 **DCE** device.

See the *Serial Port* section in the *Operation* chapter for more information.

1.3.2 SIM card slot

Use this slot to insert the GSM SIM card. Insert the SIM card as described on the next figure. Use a thin object like a coin to get the SIM card fully inserted until it clicks.



The SIM gets locked when it clicks. A click is only possible with the correct orientation.

To remove the SIM card push it with a thin object until it clicks.

1.3.3 LEDs

Four leds are provided:

- ON LINE: Green.
- FIX: Yellow.
- SIGNAL: Orange.
- POWER: Red.

See the *LEDs* section on the *Operation* chapter for more information.

1.3.4 AUDIO jack

The audio connector is designed to connect a non-balanced hands-free audio system.

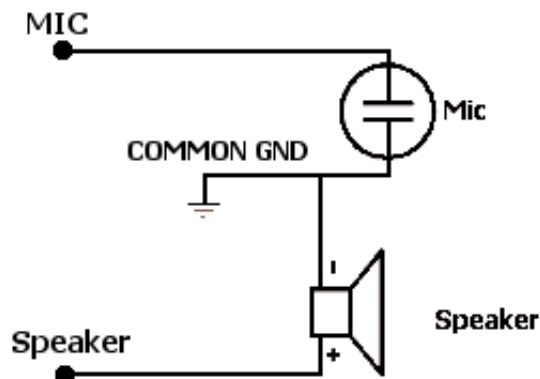
See the *Specifications* section for information on the type of speaker and microphone that can be used.

The connector is designed to use a 2.5mm stereo plug with the following configuration:



2.5mm STEREO PLUG

A non-balanced speaker-microphone connection must be as follows:



AUDIO CONNECTION

Most popular cellular phones' hands-free that use a 2.5mm stereo plug are compatible with this design.

1.4 Back side description



1.4.1 GSM Antenna connector

This is a SMA (Sub Miniature A) connector with a female *center contact*. Use this connector for the GSM Antenna provided with the unit.

1.4.2 GPS Antenna connector

This is a SMA (Sub Miniature A) connector with a male *center contact*⁵. Use this connector for the GPS Antenna supplied with the unit.

The reverse-polarity condition of the connector and of the GPS antenna connector assures that the GSM and GPS antenna will not get erroneously interchanged.

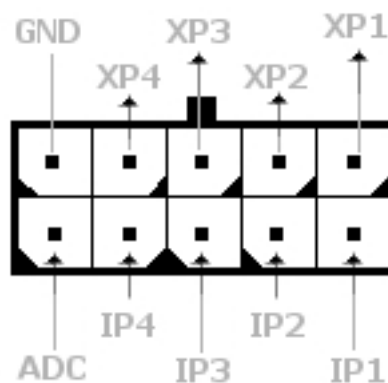
1.4.3 I/O molex-type connector

This male molex-type connector is used for the following signals:

- 4 Discrete inputs.
- 4 Discrete outputs.
- Voltage Input for the ADC converter.
- Ground.

The pin-out of these signals is:

⁵ Also called reverse polarity connector



I/O MOLEX

XP makes reference to Outputs, IP to Inputs. For information on inputs, outputs and ADC see the *Operation* chapter.

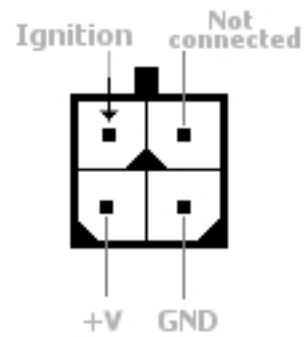
The I/O Harness (female molex-type receptacle) supplied with the unit is configured as follows:

| Signal | Color |
|--------|------------------|
| IP1 | White and brown |
| IP2 | White and red |
| IP3 | White and orange |
| IP4 | White and yellow |
| ADC | White |
| XP1 | Blue and brown |
| XP2 | Blue and red |
| XP3 | Blue and orange |
| XP4 | Blue and yellow |
| GND | Black |

1.4.4 Power/ignition molex-type connector

This male molex-type connector is used for the unit's main power connection (vehicle's battery) and for the ignition detector. The pin-out of these signals is:

1.4. BACK SIDE DESCRIPTION



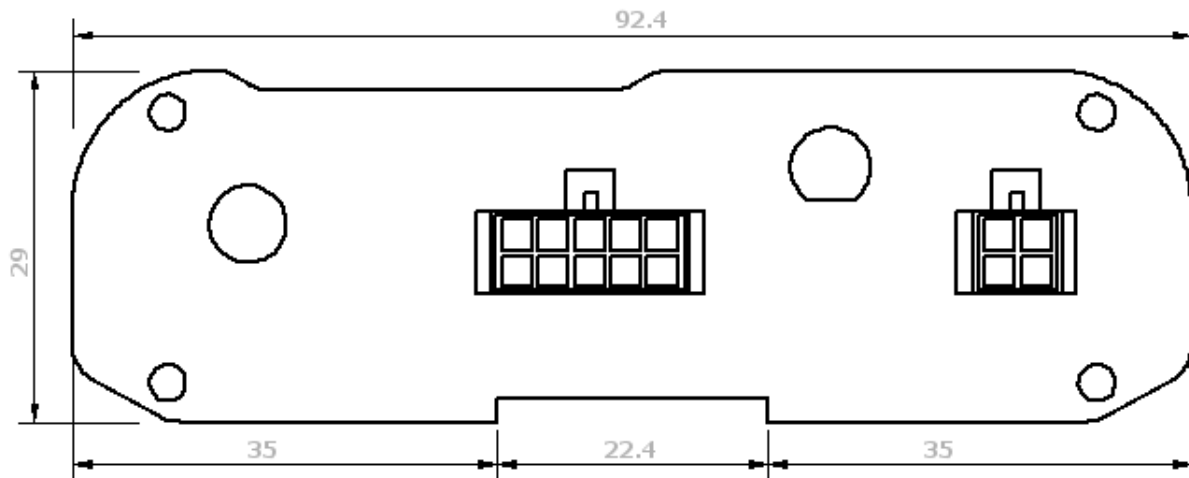
POWER MOLEX

The Power Harness (female molex-type receptacle) supplied with the unit is configured as follows:

| Signal | Color |
|----------------|--------|
| +V | Red |
| GND | Black |
| Ignition (F00) | Yellow |

1.5 Specifications

1.5.1 Dimensions



DIMENSIONS

Values shown in millimeters. The depth of the unit is 127mm.

1.5.2 Environment

- Operating: -20°C to $+55^{\circ}\text{C}$
- Storage: -30°C to $+85^{\circ}\text{C}$
- Humidity: Up to 95% non-condensing.

1.5.3 Power

- DC Voltage : 8V - 32 V
- Current consumption:
 - With internal battery at full charge (IDLE): 60mA @ 12V.
 - With internal battery at zero charge (IDLE): 600mA @ 12V.
 - Without internal battery (IDLE): 60mA @ 12V.
 - On sleep mode: 1mA @ 12V.
- Reverse voltage polarity protection.
- Thermal shutdown and current limit protection.

1.5.4 Inputs/Ignition

- Input impedance: 50 Kohms.
- Internal Pull-up: 50 Kohms.
- Sampling rate: 3 s/sec.

Inputs' detection:

| Logical State | Electrical State |
|---------------|---------------------|
| Active | 0V to 1V |
| Inactive | 2.9V to 32V or Open |

Ignition detection:

| State | Voltage range |
|--------------|--------------------|
| Ignition ON | 5.8V to 32V |
| Ignition OFF | 0V to 4.8V or Open |

1.5.5 Outputs

- Open Drain
- Continuous current capacity: 2A.
- Maximum instantaneous current (< 1 sec.): 10A.
- Maximum switching voltage: 30V.
- Maximum repetition rate: 1 sec.

Logical/Electrical state:

| Logical State | Electrical state |
|----------------|---------------------------------------|
| Active (high) | 0V |
| Inactive (low) | Open or the pull-up voltage (max 30V) |

1.5.6 Analog To Digital Converter

- Voltage range: 0 V to 32 V.
- Input impedance: 22 Kohms.
- Resolution: 10 bits.
- Sampling rate: 3s/sec averaged on a 10 seconds interval.

1.5.7 Audio

- Supports non-balanced mic-speaker connection.

- 2.5mm Stereo audio jack.
- Speaker impedance: 32ohms.
- Microphone: Electret type.

1.5.8 GSM/GPRS module

- Frequencies: 850/1900 or 900/1800 Mhz.
- Full duplex communication.
- Automatic start up.
- Antenna Impedance: 50 Ohms.
- SIM Card: 3 Volts.
- SIM Card PIN: Programmed by user one time, automatically introduced onward. PIN saved on *Non-Volatile* memory.
- Receiver parameters:
 - GSM850 Reference Sensitivity = -104 dBm Static and TUNHigh.
 - E-GSM900 Reference Sensitivity = -104 dBm Static and TUNHigh.
 - DCS1800 Reference Sensitivity = -102 dBm Static and TUNHigh.
 - PCS1900 Reference Sensitivity = -102 dBm Static and TUNHigh.
 - Selectivity @ 200 kHz : > +9 dBc
 - Selectivity @ 400 kHz : > +41 dBc
 - Linear dynamic range: 63 dB
 - Co-channel rejection : >= 9 dBc
- Transmitter parameters:
 - Maximum output power (EGSM and GSM850): 33 dBm +/- 2dB at ambient temperature
 - Maximum output power (GSM1800 and PCS1900): 30 dBm +/- 2dB at ambient temperature
 - Minimum output power (EGSM and GSM850): 5 dBm +/- 5dB at ambient temperature
 - Minimum output power (GSM1800 and PCS1900): 0 dBm +/- 5dB at ambient temperature

1.5.9 GPS module

Units with firmware version 1.05:

- Sensitivity: -152 dBm Tracking, -142 dBm Acquisition.
- Protocol: TAIP(ASCII).
- Frequency: L1 type (1575.42 MHz). C/A code.
- Channels: 12 channel simultaneous operation.
- Update rate: 1Hz.
- Accuracy:
 - Horizontal: <3 meters (50%), <8 meters (90%)
 - Altitude: <10 meters (50%), <16 meters (90%)
 - Velocity: 0.06 m/sec.
 - PPS: +/-50 nanoseconds.
- Acquisition:
 - Reacquisition: 2 sec.
 - Hot Start: 9 sec.
 - Warm Start: 35 sec.
 - Cold Start (TTFF): 39 sec.
 - Out of the box: 41 sec.

Units with firmware version 2.01:

- Supports SBAS (WAAS, EGNOS).
- Sensitivity: -150 dBm Tracking, -142 dBm Acquisition.
- Protocol: TAIP(ASCII).
- Frequency: L1 type (1575.42 MHz). C/A code.
- Channels: 12 channel simultaneous operation.
- Update rate: 1Hz.
- Accuracy:
 - Horizontal: <2.5 meters (50%), <5 meters (90%)
 - [SBAS]: <2 meters (50%), <4 meters (90%)
 - Altitude: <5 meters (50%), <8 meters (90%)
 - [SBAS]: <3 meters (50%), <5 meters (90%)
 - Velocity: 0.06 m/sec.
 - PPS: +/-100 nanoseconds RMS.
- Acquisition:
 - Reacquisition: 2 sec.

1.5. SPECIFICATIONS

Hot Start: 3.1 sec.
Warm Start: 35.4 sec.
Cold Start (TTFF): 39.4 sec.
Out of the box: 41 sec.

Units with firmware version 3.02:

- Supports SBAS (WAAS, EGNOS).
- Sensitivity: -160 dBm Tracking,
-142 dBm Acquisition(Standard Sensitivity Mode),
-148 dBm Acquisition(Hot Start with ephemeris, otherwise -146 dBm. High Sensitivity Mode).
- Protocol: TAIP(ASCII).
- Frequency: L1 type (1575.42 MHz). C/A code.
- Channels: 12 channel simultaneous operation.
- Update rate: 1Hz.
- Accuracy:
Horizontal: <2.5 meters (50%), <5 meters (90%)
[SBAS]: <2 meters (50%), <4 meters (90%)

Altitude: <5 meters (50%), <8 meters (90%)
[SBAS]: <3 meters (50%), <5 meters (90%)

Velocity: 0.06 m/sec.
PPS: +/-100 nanoseconds RMS.
- Acquisition:
Reacquisition: 2 sec.
Hot Start: 3 sec.
Hot Start w/o battery back-up: 8 sec (Ephemeris is not older than 4h).
Warm Start: 35 sec.
Cold Start (TTFF): 38 sec.
Out of the box: 41 sec.

To consult or update the firmware version of your *Antares SBTM* GPS module, please contact *Digital Communications TechnologiesTM*. Updates are provided at no cost.

1.5.10 GSM antenna connector

SMA (Sub Miniature A) connector with a female *center contact*. Use this connector for the GSM Antenna provided with the unit.

1.5. SPECIFICATIONS

- 50 ohms impedance.

1.5.11 GPS antenna connector

SMA (Sub Miniature A) connector with a male *center contact*⁶.

- 50 ohms impedance.

⁶ Also called reverse polarity connector

2 Installation

The *Antares SBTM* can be installed in any location of any type of vehicle¹ as long as some environmental conditions are met:

- No exposure to water.
- No direct exposure to direct sun light.
- Away from excessive heat sources like the motor or the exhaust's path.
- Away from excessive cold sources like a truck's refrigerator or AC system.
- Not attached to a highly vibrating structure.

The unit's location/position can be such that it remains hidden. The *LEDs* indicators do not have to be visible but it is recommended some access to them for failure/diagnostics situations. The same recommendation holds for physical access to the unit's serial port.

2.1 Power Supply

The unit's power cables can be directly connected to the vehicle's battery (12 or 24 volts). The maximum voltage the unit can take is 32V.

When using the unit outside a vehicle use a 12Volts DC adapter that supplies a minimum current of 800mA.

2.1.1 Vehicles with a main power switch

When the vehicle has a main power switch to cut/restore the battery voltage, some recommendations have to be followed:

- If the switch disconnects the positive voltage of the vehicle's battery, the *Antares SBTM* can be connected before or after the switch. When connected before it will keep on receiving the vehicle's power whenever the switch is off. If it is connected after the switch, the unit will run with its optional back up battery whenever the switch is turned off.

¹ See the warning about vehicles that use a main switch for cutting/restoring the negative terminal of the vehicle's battery described on the *Power supply* section.

2.1. POWER SUPPLY

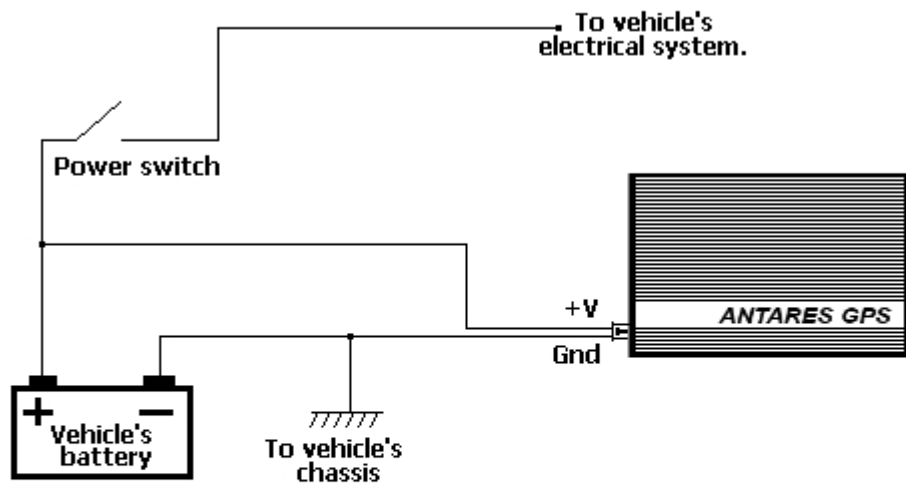
- If the switch disconnects the negative voltage of the vehicle's battery, the *Antares SBTM* power CAN NOT BE TAKEN BEFORE THE SWITCH. Doing so will make all of the unit's connections to ground like panic buttons and its chassis serve as current path canceling the vehicle's main power switch thus making large currents circulate through the unit. For this configuration it is mandatory taking the unit's power after the switch, making the unit work with its back-up battery every time the main power switch is off.

Warning:

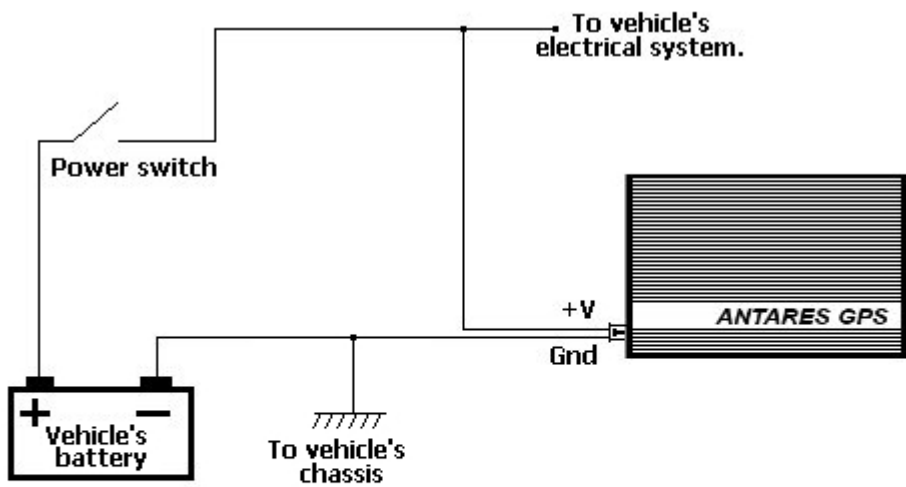
When the vehicle uses a *switch for the negative line* that goes between the vehicle's battery and the vehicle's chassis DO NOT CONNECT THE UNIT'S GND TO THE BATTERY'S NEGATIVE. This could seriously damage the unit. Connect the unit's GND to the vehicle's chassis (after the switch) so the unit runs on its optional back-up battery while the switch is off.

See the next figures for a better understanding of connecting the unit when the vehicle uses a main power switch:

2.1. POWER SUPPLY

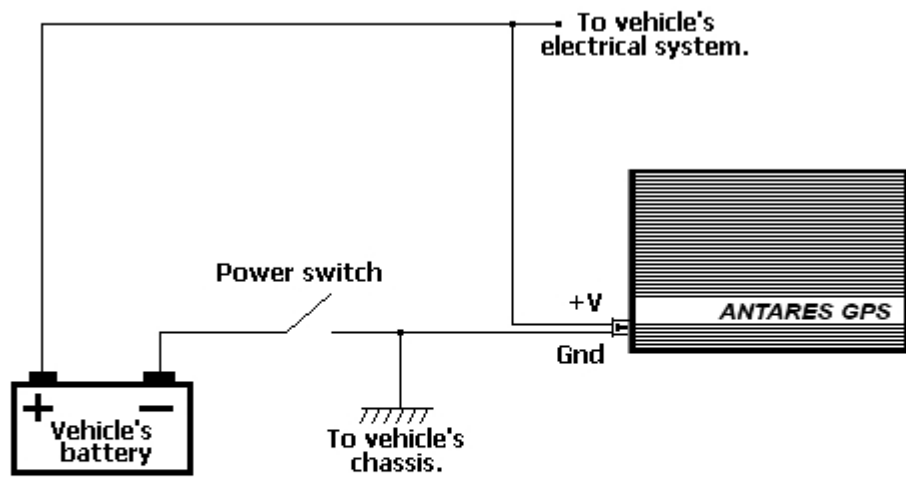


Power switch at positive and
Antares connected before:
OK.

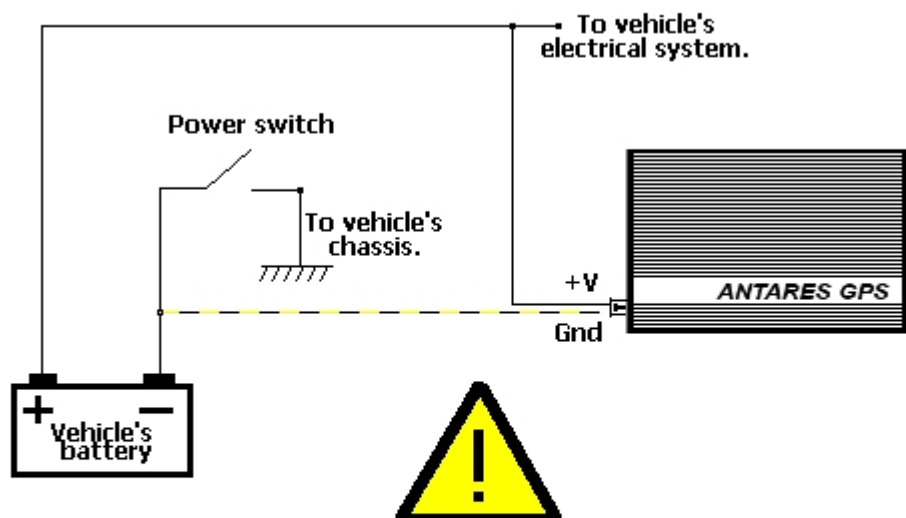


Power switch at positive and
Antares connected after:
OK.

2.1. POWER SUPPLY



Power switch at negative and
Antares connected after:
OK.



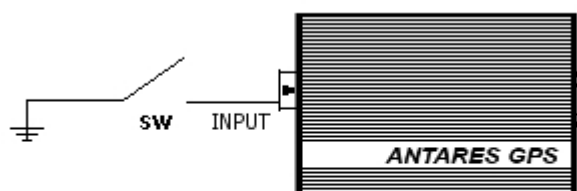
Power switch at negative and
Antares connected before:
WRONG.

2.2 Inputs detection

For the general purpose inputs the electrical conditions are as follows:

| Logical State | Electrical State |
|---------------|---------------------|
| Active | 0V to 1V |
| Inactive | 2.9V to 32V or Open |

A typical input configuration consists of the input connected through a switch to GND. This makes the input float whenever the switch is off indicating to the unit that the input is open, and makes the input go to 0V when the switch is close indicating an ON condition to the unit.



INPUT DETECT

A voltage detection can be used too:

- Any voltage above 2.9V on the input will be indicated as OFF (inactive).
- Any voltage below 1V on the input will be indicated as ON (active).

It is normal to see a 3.4V(approx.) voltage on the *Antares SBTM* Inputs when nothing is connected to it. This voltage is set on purpose through a pull-up circuit (50kOhm resistors) in order to fix a inactive state when there is nothing connected to the Inputs. The pull-up circuit also allows to connect an open drain or open collector output of a device directly to *Antares SBTM*. The Inputs are connected internally to several protection circuits, including over voltage protection among others. The maximum input voltage is 32V.

2.3 Ignition detection

The electrical conditions for the ignition input are:

| State | Voltage range |
|--------------|--------------------|
| Ignition ON | 5.8V to 32V |
| Ignition OFF | 0V to 4.8V or Open |

2.4. OUTPUTS

This detection is different than inputs'. The ignition detection circuit is different than inputs': Any voltage above 5.8V on the ignition input will be detected as ignition ON. Anything below 4.8 will be detected as ignition OFF.

Under this conditions, this input is designed to be connected to the ignition's key position that closes the circuit between the battery's positive voltage and the vehicle's electrical system. It should not be connected to the *start* position which gives energy to the vehicle's start engine as this position is ON only for a short period of time. The unit's ignition sense can be directly connected to the electrical end of this position. See the *Connection Diagrams* section for an illustration.

It is normal to see a 3.4V(approx.) voltage on the *Antares SBTM* ignition input when nothing is conected to them. This voltage is set on purpose through a pull-up circuit (50kOhm resistors) in order to fix a inactive state when there is nothing connected to the ignition input. The ignition input is connected internally to several protection circuits, including over voltage protection among others.

2.4 Outputs

The unit has 4 discrete outputs located on the I/O molex-type connector.

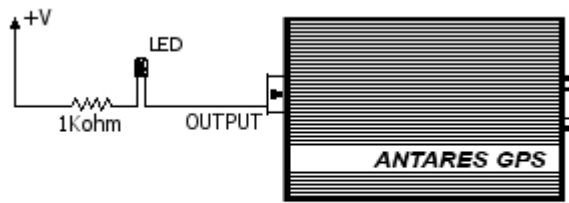
The outputs are *Open-Drain* type with no internal pull-up resistor. Meaning that the user has to provide a pull-up resistor to any positive voltage (30V max.) to detect an inactive output by voltage. Each output can drive a continuous current of 2A.

The electrical conditions are:

| Logical State | Electrical state |
|---------------|---------------------------------------|
| Active | 0V |
| Inactive | Open or the pull-up voltage (max 30V) |

If the output is used for cutting/restoring GND, a direct connection can be used. For example when driving a LED:

2.4. OUTPUTS



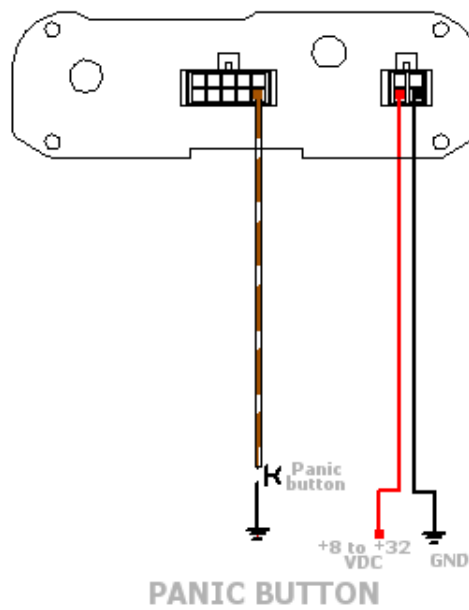
LED DRIVE

If the output is used for cutting/restoring a positive voltage on a high current device, like for example the vehicle's ignition wire, an external device like a relay or high current transistor has to be used. See the *Connection diagrams* for an illustration.

2.5 Connection diagrams

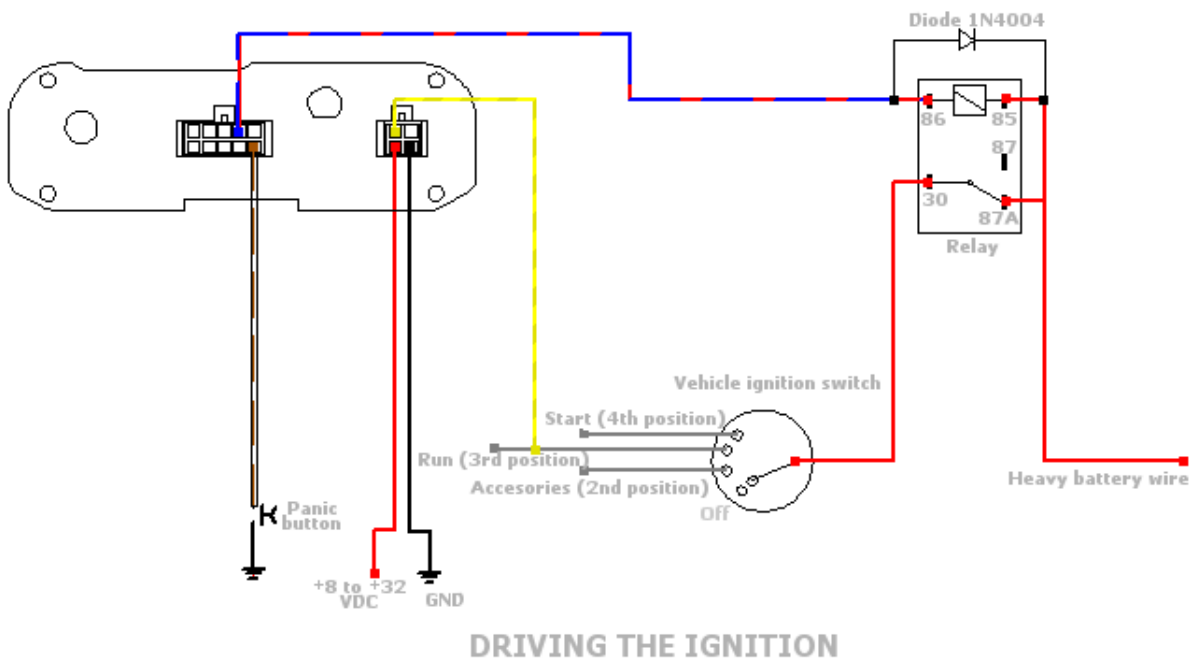
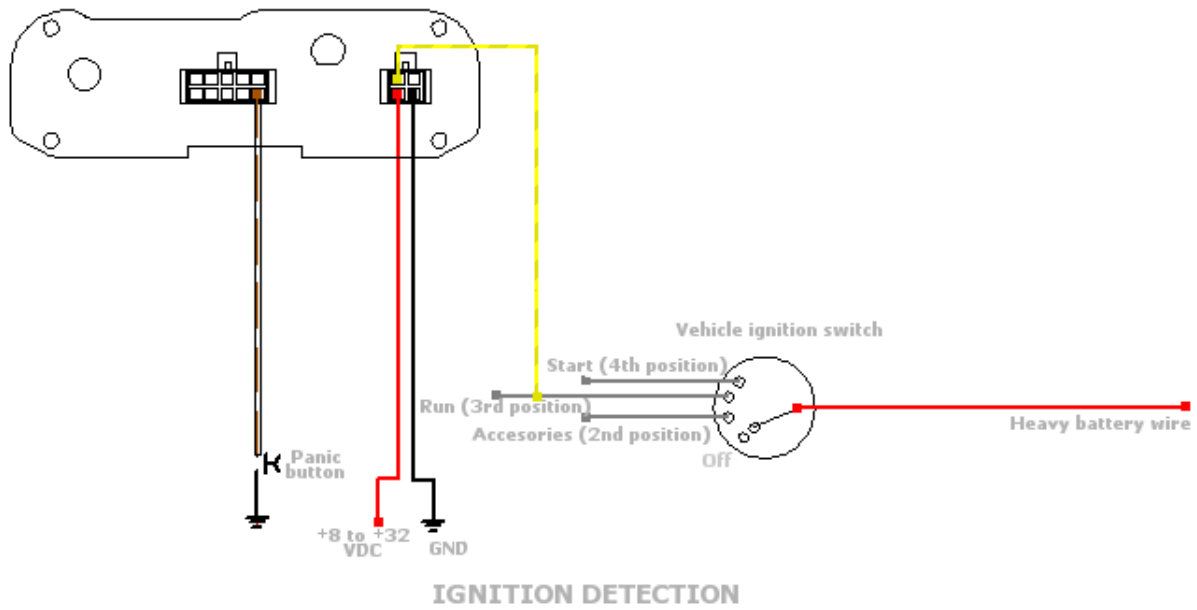
The following illustrations show:

- Panic Button detection.
- Ignition sense.
- Engine turn off².



² Shutting and engine off without knowing a vehicle's state is not advised. See the *Operation* chapter for more information.

2.5. CONNECTION DIAGRAMS



3 Operation

The *Antares SBTM* operates on an automatic basis according to an user-defined configuration which can be modified at any time locally or remotely using the GSM/GPRS network. The unit's operation includes actions/reports based on interaction with its surroundings like remotely-given commands, input sensing, accessories' messages, etc.

The unit does not requires any starting command/action to start working: Once a valid configuration is loaded the unit is ready to work.

Note:

Follow the *essential* configuration parameters marked with an “*” on the *Configuration* chapter in order to have the minimal operational parameters.

Many configuration scenarios are possible, the most common one for the unit's operation is having it reporting a periodic *status event* to any IP server¹ according to a *Time And Distance*² or *Time-only* criteria and having the unit report other sort of events as an exception to this normal status event reporting. Such an example is configuring the unit to report an input change to the same IP address and to a phone number via SMS whenever a driver presses a panic or assistance button.

For information on how to configure the unit refer to the next section and to the *Configuration* chapter.

¹ Running an AVL server.

² A *Time And Distance* criteria is preferred over a *Time-only* criteria

3.1 Serial Port

The unit's RS-232 serial port can serve one of two non simultaneous purposes:

1. Make a user or software interact with the unit's TAIP console for configuration and operational purposes. In this mode only printable ASCII characters are used to communicate with the unit.
2. Have the unit exchange any binary messages with accessories like MDTs or PDA-like devices that can communicate over RS-232. In this mode any binary data except an user-defined escape value may be used.

The unit's default setting is to work as described on the first option. That is it, the serial port is ready to exchange TAIP messages with the unit's TAIP console unless the user switches to the second mode. In the second mode the unit will not listen to TAIP commands and the procedure to set it back to the TAIP console is receiving a pre-configured escape character or a string sent as a single package. The MT TAIP messages controls the serial's port mode, see the TAIP reference and the *Configuration* chapter for more information.

The serial port works at 9600bps with 8 data bits, no parity and one stop bit (9600,8N1). It does not use any flow control method and all of the RS-232 communication lines on the DB9 connector are used. The *Antares SBTM* works as a DCE device, meaning that it is connected to a PC with a one-one cable.

The serial port works with pure RS-232 hardware as well as with USB to RS-232 converters.

Note:

When using the unit's serial port for the first time, the >QVR< TAIP command can be used to test the communication path. The unit should respond with its firmware version on a message like this: >RVR; Antares GPS 05.30;ID=0000<.

Once you are able to communicate you can start interacting with the unit's TAIP console, refer to the *TAIP console* section on this chapter for more information.

3.2 LEDs

The unit's four LEDs are functional all the time.

3.2.1 Power (Red)

This LED is solid ON whenever the unit is functional, OFF when it has no main or back-up power³. Any time the LED is blinking the unit is in a temporal state where some features are not available. These temporal states may be reached when:

- Initializing: This state lasts 15 seconds and is reached any time the unit is recovering from a non-power situation or recovering from a previous system reset. If the unit remains in this state for a longer time you may have a hardware problem. In this state the unit's TAIP console will not respond to commands.
- Signing-off the GSM/GPRS network: Before a system reset the unit signs off the network, this procedure takes from 3 to 10 seconds. This procedure is also done before entering *sleep* mode.
- On *sleep* mode: The LED blinks very shortly on a 4 seconds basis.

There is one exception for this type of situations when the LED is blinking:

- The *Power* LED is blinking at unison with the *Signal* LED: This does not indicate a temporal lack of functionality but SIM card's initialization. This is a temporary state that should not last more than 6 seconds and it may only happen after a system reset. If it lasts longer you may have a hardware problem or a defective SIM card.

3.2.2 Signal (Orange)

As long as the Power LED is solid ON, this LED indicates the GSM registration status in the following way:

- Solid: The unit is **Not Registered** on the GSM network.
- Blinking: The unit is **Registered** on the GSM network.
- Off: The unit is **Registered** with a very poor signal.

When the unit is GSM-registered, the *Signal* LED is either blinking or completely off. By counting the number of blinks before a pause the user can have an estimate of the *Received Signal Strength*. See the table below.

³ It may be also OFF when in an special technical support mode.

| Blinks | RSSI | RSS [(-)dBm] | |
|--------------|----------|--------------|-----------|
| 0 (off) | 0 to 7 | 113 - 99 | Very poor |
| 1 | 8 to 13 | 97 - 87 | Poor |
| 2 | 14 to 17 | 85 - 79 | Fair |
| 3 | 18 to 20 | 77 - 73 | Fair |
| 4 | 21 to 24 | 71 - 65 | Good |
| 5 (no pause) | 25 to 31 | 63 - 51 | Excellent |

3.2.3 Fix (Yellow)

As long as the *Power* LED is solid ON, this LED gives information on the GPS receiver status. There are two possible states:

- ON: The GPS unit is doing fixes. This indicates a well placed GPS antenna with sky view.
- OFF: The GPS antenna is connected but the unit is not doing fixes. This could happen even if the antenna is well placed, but if this is the case, the situation should not be permanent. In this situation the GPS messages transmitted by the unit may be of lower GPS quality. Check the GPS antenna location and/or type if the situation persists.

3.2.4 On line (Green)

As long as the *Power* LED is solid ON, this LED gives information on the GPRS session state:

- OFF: The unit is **Not Registered** on the GPRS network.
- ON: The GPRS session is *up and ready*. When using TCP it also indicates that all of the TCP connections with all of the IP-type *Destinations* are ready.
- Blinking: The GPRS session is *up and ready* but at least one of the TCP sockets associated with IP-type destinations is not open.

When using UDP the LED is either ON or OFF.

Note:

When an APN is set the unit will always try to start and maintain a GPRS session so the normal state of the *On line* LED when an APN is set is either ON or Blinking.

3.3 Inputs/Ignition

The unit has 4 general purpose discrete-inputs located on the main molex connector and a vehicle ignition detector located on the main power molex connector.

For the general purpose inputs the electrical conditions are as follows:

| Logical State | Electrical State |
|---------------|---------------------|
| Active | 0V to 1V |
| Inactive | 2.9V to 32V or Open |

For the ignition input:

| State | Voltage range |
|--------------|--------------------|
| Ignition ON | 5.8V to 32V |
| Ignition OFF | 0V to 4.8V or Open |

The Inputs' state can be consulted locally or remotely at any time with the TAIP console.

Inputs are used to create events' triggers on the *Event Machine* thus generating reports depending on the inputs' state/changes.

To create input dependent events and to consult inputs' state the *Event Machine's* IP signals are used. These signals are *true* whenever the corresponding input is *true*. For information on signals consult the *Event Machine* section on the *Configuration* chapter.

The corresponding signals are:

| Input | Signal name | Old signal name |
|----------|-------------|-----------------|
| Input 1 | IP1 | G00 |
| Input 2 | IP2 | G01 |
| Input 3 | IP3 | G02 |
| Input 4 | IP4 | G03 |
| Ignition | F00 | F00 |

Inputs can be monitored with the SS TAIP message.

Example To consult the state of Input 3:

Using the TAIP console send to the unit:

```
>QSSIP3<
```

For an active input (i.e. input 3 at GND) the unit returns:

```
>RSSIP31<
```

3.4. OUTPUTS

and when the input is not active (i.e. input 3 at any voltage⁴ or disconnected):

>RSSIP30<

3.4 Outputs

The unit has 4 discrete outputs located on the main molex connector. Given their 2A current capacity they can drive a wide range of loads.

The outputs are of *Open Drain* type with no pull-up resistor. Meaning that the user has to provide a pull-up resistor to any positive voltage (30V max.) to detect an inactive output by voltage. Each output can drive a continuous current of 2A.

The electrical conditions are:

| Logical State | Electrical state |
|---------------|---------------------------------------|
| Active | 0V |
| Inactive | Open or the pull-up voltage (max 30V) |

Outputs may be driven locally or remotely using the TAIP console or the unit can be configured to automatically take outputs' actions depending on different situations. The *Event Machine's* signals associated to outputs are:

| Output | Signal name | Old signal name ⁵ |
|----------|-------------|------------------------------|
| Output 1 | XP1 | G04 |
| Output 2 | XP2 | G05 |
| Output 3 | XP3 | G06 |
| Output 4 | XP4 | G07 |

Outputs activation and deactivation is controlled by the SS TAIP message.

Example To activate output 3 send to the unit:

>SSSXP31<

To query its status:

>QSSXP3<

and the unit should return:

>RSSXP31<

To deactivate the output:

⁴ The maximum supported voltage for inputs is 32 volts.

>SSSXP30<

Example Making the *Antares SBTM* automatically drive an output:

Use the **SS** message as an user-action on an event definition, see the *Event Machine* section on the *Configuration* chapter and the **ED** message on the TAIP reference:

>SED20NV0;R05-;ACT=SSSXP21<

This will make the unit automatically set the Output 2 high when it leaves region R05. See the **GR** message on the TAIP reference for information on how to create regions.

3.5 Analog to Digital Converter

The *Antares SBTM* has one ADC whose input voltage is located on the I/O molex-type connector.

The input voltage range is 0V to 32V. The ADC value computed by the unit is an average value of samples taken at a 10 seconds period, so you will not correctly get a voltage pulse that last a shorter time. For any change to be accurately read the 10 seconds has to pass in order for the average computation to get stable.

This reading method gives the converter the ability to filter rapid changes or voltage swings that are undesirable when the voltage is proportional to some physical variables prone to this behavior.

The actual computed value in millivolts may be consulted at any time with the **XAAC** message, see the TAIP reference for more information.

Example Query the actual computed value:

>QXAAC<

For an ADC computed value of 23.344 volts (72% OF 32V)the unit responds:

>RXAAC23344P072<

Sending reports whenever the ADC reaches some value or whenever it falls from another value⁶ is possible by using the *Event Machine*

⁶ Up to 5 ADC thresholds may be defined with the **XAGA** message.

3.6. BACK-UP BATTERY

with the D signals which are configured with the **XAGA** message.

Example Have the unit send event code 23 whenever the ADC input voltage goes beyond 20 volts and code 21 whenever the voltage falls below 12 volts:

Create the 2 ADC thresholds:

```
>SXAGA01V12000<
```

```
>SXAGA02V20000<
```

Create the events associated with signals D01 and D02:

```
>SED21XV0;D01-<
```

```
>SED23XV0;D02+<
```

These events will send the report to the serial port. See the *Event Machine* section on the *Configuration* chapter for more information.

3.6 Back-up Battery

The *Antares SBTM* offers the possibility of a built-in back-up Lithium-Polimer battery to be used whenever the main power source (the vehicle's battery) is lost or sabotaged.

Warning:

Not all of the units come with the built-in back-up battery, contact *Digital Communications TechnologiesTM* or your dealer and ask for information.

The duration of the back-up power depends on many factors like:

- Frequency of the unit's reporting.
- Network conditions: Poor network signal demands more power from the unit.
- Battery condition: At full charge, mid-charge, etc.
- Unit working on the *sleep* or *normal* mode.

Having the back-up battery at full charge on good network conditions the measured durations are:

- 10 hours at full rate GPRS transmission: Having the unit reporting to an IP-type destination on a 10 seconds basis.

3.6. BACK-UP BATTERY

- 10 days when in *sleep* mode: Only inputs' detection is available and an optional wake-up interval is available. This last wake-up option was not used on the battery test.

The battery level measurement is done in a similar way as the unit's *Analog to Digital Converter* please refer to that section for some notes.

Note:

The battery state is not an instant value of the battery level, instead it gives an average value computed every 10 seconds.

The actual computed value may be consulted at any time with the XABS message, see the TAIP reference for more information.

Example Query the actual battery status:
>QXABS<

For a back-up battery at 3.98 volts (72% of charge):
>RXABS13980P072<

The first “1” is indicating that the main power source is ON.

To change a reporting schedule and/or report when the main power source is lost or damaged the F13 signal is used. See the *Event Machine* section on the *Configuration* chapter for more information.

Example Send event code 44 whenever the main power source gets disconnected:
>SED44XV0;F13-<

Sending reports whenever the battery level reaches some value or whenever it falls from another value⁷ is possible by using the *Event Machine* with the B signals which are configured with the XAGB message.

Example Have the unit send event code 19 whenever the battery level gets higher than 90% and code 20 whenever the voltage falls below 40%:

Create the 2 battery level thresholds:
>SXAGB03P00090<

⁷ Up to 5 battery level thresholds may be defined with the XAGB message.

3.7. SLEEP MODE

>SXAGB04P00040<

Create the events associated with signals B03 and B04:

>SED19XV0;B03+<

>SED20XV0;B04-<

These events send the report to the serial port. See the *Event Machine* section on the *Configuration* chapter for more information.

3.7 Sleep Mode

Refer to the XAPM TAIP message for more information on sleep mode.

3.8 Over The Air

Interacting with the unit remotely (Over the Air) is not different than doing it locally. The unit's configuration and operation is controlled by means of its TAIP console which is a command-response mechanism that allows to change configurations, operate outputs and consult the state of the unit. This console can be accessed over the air via IP and/or via SMS messages and locally over the serial port.

In order for the TAIP console to be used remotely a *Destination* (or several of them) has to be defined so the unit knows who is authorized to interact with it. *Destinations* are discussed on the *Configuration* chapter and are configured with the XADP TAIP message, refer to those sections for information on how to create and configure *Destinations*.

Note:

TAIP queries that generate multiple answers are not supported over the air. For example sending >QED< without specifying an event's ID is only supported over the serial port.

3.8.1 Via IP hosts (GPRS)

To interact with the unit via IP hosts, the corresponding IP address/name⁸ and TCP/UDP port has to be created on any of the unit's *Destinations*' space. When using TCP the unit functions as a TCP client which always initiates the connection to the remote IP

⁸ *Antares SBTM* supports IP numbers as well as domain names.

host which acts as a TCP server. Once the connection is established the TCP server can send any TAIP command/query to the unit using the established connection.

When using UDP nobody initiates a connection. In this case the unit responds TAIP queries sent as UDP datagrams from a remote IP address/name defined on an UDP-IP-type *Destination*. There are two mechanisms for interrogating the unit via UDP datagrams: Using the unit's UDP server port and/or using the unit's UDP-origin port. The latter is by default dynamically generated and changed by the unit as datagrams are sent; but using the TAIP command **XAU0** this value can be fixed.

- UDP Server Port: Set a value from 1 to 65535 with the command **XAUP** so the unit always listens to UDP datagrams containing TAIP commands on a fixed, always-available port. Notice that the UDP server port solution only works when both the server sending the TAIP query and the *Antares SBTM* are located on the same network (Virtual or Real). This is not the case on the majority of situations where the unit has Internet access.
- UDP Origin Port: The *Antares SBTM* always listens for TAIP queries sent over UDP datagrams to its UDP origin port. The UDP origin port is created whenever the unit sends an UDP datagram to a remote peer; so, if the unit has never sent a datagram, it will not create and hence listen on this port. Also notice that the UDP origin port the unit creates is attached to the remote peer's address and port (socket); so, an UDP origin port only listens to UDP datagrams coming from the peer to which the last report was sent. In fact there is more than one UDP origin port, there is at least one for each peer the unit has sent reports to. The UDP origin port is the only solution for interrogating a unit that is behind a NAT, which is the case on the majority of units that have Internet access. Notice that the UDP origin port can be set to a fixed value; this is only useful on units which work on the same network as the AVL server(s) (remote peer(s)) because when units work behind a NAT, the origin port seen by the remote peer is always set by NAT rules, no matter *Antares SBTM* uses a fixed value.

The unit may accept TAIP queries coming from any of the 10 possible UDP origin ports and also from the UDP server port. The server port, which is only useful when the unit works on the same network as the AVL server, is not enabled by default.

Server address validation

When working with TCP, a server sending TAIP queries to *Antares SBTM* is validated through the TCP connection which is always initiated by the unit.

For incoming UDP datagrams there are two mechanism the unit uses to validate the remote peer (AVL server).

1. First, the unit tries to match the remote peer address with the UDP Network set with the **XAUN** command. If the remote peer address does not match the UDP network, or if the UDP network is not defined (default state), the second validation mechanism is performed.
2. The second validation is done according to the *Destination Points* list. If the server address is found on the list the server is validated an the TAIP command is accepted (as long as the corresponding Destination Point has the TAIP console access enabled (default state)).

The *Destination Points* list can be set and consulted with the **XADP** command.

There are some considerations to be taken when the unit is working behind a NAT (very common when the unit has Internet access):

If the *Destination* defined for TAIP console access is also the AVL server which listens for the unit's periodic and exceptional reports there has to be some reporting criteria that guarantees that the unit will not be silent for a period of time longer than the NAT's port expiration time. If the unit is silent for a longer period, the NAT server on the cellular carrier's side will silently close an active TCP connection or eliminate any UDP port translation making it impossible for the AVL server to send queries to the unit until the unit reports again. In a similar situation when the TAIP-console access is granted to an IP host other than the AVL server which recollects the unit's reports, a periodic time-only criteria has to be defined to generate a report to the IP host so it can consult the unit at any time.

For TCP this time is around 1 hour and for UDP is around 5 minutes but this is dependent on the cellular operator and may be different in your case.

This situation has two possible workarounds:

1. From a SMS *Destination* (mobile number) that has TAIP console access, send a SMS with a command that instructs the unit to send a “hello-type” report to the IP host which is going to interact with the unit. Thus this SMS will have to be sent only when the communication via IP is lost due to a NAT expiration time. This eliminates the need of a keep-alive for TCP or UDP report and depending on the SMS’ charging value it may lead to a cost effective solution for having the unit’s TAIP console remotely available all the time. For a “hello-type” report see the **XACT** TAIP command.
2. Have the units work on a private network with the IP hosts so there is no NAT involved.

For more information on the NAT problem refer to the *Internet and NATs* section of this chapter.

3.8.2 Via SMS (GSM)

To interact with the unit via SMS messages, a *Destination* holding the sender’s Telephone Number has to be defined. As Telephone Number destinations are also used for voice call authorization/generation the *Destination* configuration has to be so that TAIP console access is granted. See the **XADP** TAIP message for more information.

Once the *Destination* is correctly configured the unit will respond with an SMS message to any TAIP command received via a SMS message from that destination.

Note:

When creating the *Destination* make sure of using the Telephone Number string that the unit receives from the cellular operator as generating party. Sometimes the number you use to dial is not exactly the number that appears on a phone when a SMS is generated. For example although the generating number is 123456 the cellular operator may tell the unit that the remote party is +44123456. In this case the long number including the + sign has to be entered in the unit as *Destination* or it will fail on recognizing the valid sender.

SMS messages can be sent/received even with no GPRS registration so you can have the unit working on a GSM-only basis too.

SMS messages are great for solving the NAT problem mentioned on the previous sub-section.

3.8.3 Voice (GSM)

Refer to the **XAVC**, **XAVE** commands and voice signals for more information. Refer also to the *Operation* chapter and *Examples* chapter.

3.9 TAIP console

The *Antares SBTM* TAIP console allows commands and queries to be sent to the unit by a user or software. The TAIP console is available on all of the communication channels. This means that the unit may be configured and/or consulted *locally* over the serial port or *remotely* by means of IP communication (GPRS) or SMS interaction (GSM). The console is also used to show the unit's report messages generated by the *Event Machine* (i.e EV and ET TAIP messages).

The TAIP console is based on a question/command and its answer which can be formed by multiple messages⁹. All of the messages exchanged on the console are TAIP (Trimble Ascii Interface Protocol TM) messages.

3.9.1 TAIP Message Format

All TAIP messages use printable ASCII characters. The unit can be configured to output TAIP messages in response to queries or on a scheduled basis.

Each message has the following format:

>ABB{C}{;flag=DDDD}[;flag=DDDD]<

where:

| Field | Meaning |
|-------|--|
| > | Start of a new message |
| A | Message qualifier |
| BB | Message identifier |
| C | data string |
| DDDD | Optional report flags |
| < | delimiting character |
| {x} | signifies that x can occur any number of times |
| [x] | signifies that x may optionally occur once |

⁹ Queries that result on multiple answer messages can only be used over the serial port

Start of a New Message

The > character (ASCII code 62 decimal) is used to specify the start of a new message.

Message Qualifier

A one character message qualifier is used to describe the action to be taken on the message. See the following table for the list of valid qualifiers.

| Qualifier | Action |
|-----------|--|
| Q | Query for a single sentence. |
| R | Response to a query or a scheduled report. |
| S | Configuration or <i>set</i> message. |

Message Identifier

Alphabetical characters used to identify messages. For example PV for a position-velocity message, ED for an event definition or ER for an error message. See the *TAIP reference* chapter for a full list. There are some messages which are an extension to the normal TAIP reference for which the *Message Identifier* is **XA**, these messages have a secondary identifier which is formed by the two characters following the **XA** identifier. Extended messages are also described on the *TAIP reference* chapter.

Data String

The format of a data string depends on the message qualifier and the message identifier. The formats for each messages are described on the *TAIP reference* chapter.

Optional Report Flags.

These flags are controlled by the RM message. They allow *Antares SBTM* to use certain features explained below:

- ID Flag (ID) The unit can be configured to output every message with the vehicle's ID. The default ID is set to 0000. The *Antares SBTM* will check all incoming messages for ID, if no ID is specified, the unit will accept the message. If the ID is included in messages but does not compare with the ID previously set, the message will be ignored.

Antares SBTM can be configured to use its unique IMEI number as

reporting ID. See the configuration chapter for more information.

EC Flag (Echo) When this flag is set, *Antares SBTM* will echo any correct “Set” message (**S** qualifier) with the corresponding “Response” message (**R** qualifier). For example, when setting the unit’s ID with the TAIP message:

```
>SIDTest<
```

If the EC Flag is active, *Antares SBTM* will reply with:

```
>RIDTest;ID=Test<
```

If the EC Flag is inactive, *Antares SBTM* will not return a reply to the message.

CR Flag (Carriage Return) When this flag is set, *Antares SBTM* will append a Carriage Return and Line Feed to every response or report.

ER Flag (Error) When this flag is set, *Antares SBTM* will enable error message responses. For example, when the ER flag is active, and an incorrect message is sent to *Antares SBTM*, it will reply with a error message:

```
>Qid<  
>RER00:Qid;ID=Test<
```

If the ER flag is inactive, *Antares SBTM* will not return a reply to the message.

CS Flag (Checksum) When this flag is set, *Antares SBTM* will response every message with its checksum value. When this flag is active, all the messages sent to *Antares SBTM* must contain the CS flag with the corresponding checksum value or *Antares SBTM* will reply with Error 89. For example, when the CS flag is active the response message will contain the checksum value:

```
>RRM;CS_FLAG=T;*4C<
```

If the message sent to *Antares SBTM* does not have the checksum value of the message, it will reply with Error 89:

```
>QID<  
>RER89:QID;ID=Check;*40<
```

```
>QID;*73<
>RIDCheck;ID=Check;*7B<
```

SI Flag (Command Session) The SI flag is an Optional Report flag but it is not controlled by the RM TAIP message. The Command Session ID allows an AVL server to associate each TAIP message with its corresponding answer. To use the Command Session ID, it is only required to append the Session ID message to a TAIP message sent to Antares. If a TAIP message contains the ;SI=xxxx characters, the response to this message will include those same characters. The maximum length of a Session ID message is 10 alphanumeric characters

For example, to query the *Antares SBTM* version using the Session ID, use:

```
>QVR;SI=123ABC<
```

The response will include the Session ID used:

```
>RVR ANTARES GPS 05.30;SI=123ABC;ID=test<
```

Message Delimiter

The < character signifies end of a sentence and is used as the message delimiter.

3.9.2 Reporting messages

The unit output messages when a command or query is sent or when reporting an event message generated by the *Event Machine*. These messages are either the EV or ET messages. See the TAIP reference for information on how to get the information contained on these reports.

3.9.3 Interacting

TAIP communication must be driven by a timeout-retry mechanism.

3.10 Remote host software

The *Antares SBTM* is designed to interact with Automated Vehicle Location (AVL) servers which have the ability of interpreting TAIP messages.

In this document, AVL server, AVL system, AVL host, AVL application, remote peer, all make reference to the same thing: A piece of server-type software which receives and makes some sense of the *Antares SBTM* reports. It also may have the capability of sending

queries or commands to the unit(s).

Depending on the communication mechanism the unit uses to communicate with the AVL server some requirements have to be fulfilled. See the next sections for information on this.

3.10.1 Working with TCP

In order for the unit to start connecting to an AVL server an IP-type *Destination* has to be configured first. The configuration flag for this *Destination* must indicate a TCP host. Refer to the *Destinations* section on the *Configuration* chapter.

These are the general requirements for an AVL server working with units on TCP:

1. The AVL server has to be able to work as a TCP server, as the *Antares SBTM* is always going to initiate the TCP connection acting as TCP client. Technically speaking, for this to happen the AVL server has to be able to open a TCP socket on listening mode on an available port.
2. The port number has to be the same port number used on the unit's IP-type *Destination* and it has to be clear of any Firewall, NAT and/or router restriction.
3. The AVL server has to create a new listening TCP socket whenever the actual listening socket passes to open state, this to assure that a new connection coming from another unit is accepted.
4. The server should close idle open connections after a *minimum* time of 1 hour.
5. The server should not use a keep-alive mechanism. Using the unit's keep-alive option or a periodical report event are the preferred methods. See the **XAKA** message for keep-alive options.
6. The server should have an association between unit's ID and corresponding TCP socket so any user-query can be correctly routed. There are two ways for the server to know the unit's ID:
 - (a) Extract the ID from the “;ID=” postfix of every incoming TAIP message.
 - (b) Send an ID query to an open socket for which the ID is unknown with the >QID< command.

The second option is preferred as with the first one exists the

possibility for the server to have open TCP connections with unknown IDs every time an unit opens a connection but it has nothing to report. There is a workaround for this: Having the unit send a re-connection event so every time the connection is opened the server knows who did open it. In order to do so use the **Axx** signals creating an event that sends a report every time the **A** signal associated to the corresponding IP-type *Destination* becomes true.

However it is more efficient and simple to use the second option, when possible, as it does not generate an extra report and makes it possible for the user to tell the unit to eliminate the “;ID=” postfix from every message thus saving consumption bytes.

3.10.2 Working with UDP

In order to have the unit start connecting to an AVL server, an IP-type *Destination* has to be configured first. The configuration flag for this *Destination* must indicate an UDP host. Refer to the *Destinations* section on the *Configuration* chapter. The following are the general requirements for an AVL server working with units on UDP:

1. The AVL server has to be able to listen UDP datagrams on a given port.
2. The port number has to be the same port number used on the unit's IP-type *Destination* and it has to be clear of any Firewall, NAT and/or router restriction.
3. The server should not use a keep-alive mechanism. Using a periodical report event generated by the unit is preferred.
4. The server should have an association between the unit's ID and the information from the last inbound UDP message received from the unit. This information is the last incoming message's IP and Port origins so the AVL server can route any user-query using that IP and Port as destinations.
5. An UDP destination's configuration flag can be set so the unit waits for an UDP confirmation message from the AVL server every time an event report message is sent. This UDP wait-for-ack option can be turned OFF but it is not advised as UDP will not assure the delivery of reports by itself. If wait-for-ack option is used, the AVL server has to implement this acknowledging mechanism. Otherwise and because it does not receives a confirmation message, the unit will keep on sending the same reporting message for ever. For more information continue with the next paragraph.

3.11 Reports' messages

Antares SBTM reporting criteria is based on an *Event Machine*. The *Event Machine* allows the user the creation of up to 50 events. These events can be triggered by several situations.¹¹ The *Event Machine* allows the unit to send¹² *event reporting messages* when an event occurs, and since Firmware Version 5.21 it allows the unit to report the response of any TAIP command that the user locates on the user-action section of an event definition. So, the AVL server shall expect two types of reports from the *Antares SBTM*: The *Events' Reporting Messages* and the *Responses to TAIP commands messages*.

3.11.1 Events' Reporting Messages

When a non-silent event is triggered, a reporting message is generated. The reporting message contains among the GPS state of the vehicle, the event code which triggered. There are two types of *Reporting Messages*, the ET and EV TAIP messages:

- The ET message gives information on the event code, time, date and GPS quality when the event occurred. Here it is an example of such a Reporting Message for event 38:

```
>RET381447152212;ID=EXAMPLE<
```

For detailed information on this message refer to the ET message on the *Unit's TAIP reference* chapter.

- The EV message gives information on the event code, time, date, position, velocity, heading, and GPS quality at the moment of the event occurrence. Here it is an example of such a Reporting Message for event 00:

```
>REV001447147509+2578250-0802813901519512;ID=EXAMPLE<
```

For detailed information on this message refer to the EV message on the *Unit's TAIP reference* chapter

When configuring the *Event Machine*, on each event definition, the user decides what *Reporting Message* the event shall generate. For information on events' configuration refer to the *Event Machine* section of the *Configuration chapter*.

Since Firmware Version 5.21, the unit supports the addition of information tags to the EV reporting message so that more information can be included on an event's report.

¹¹ For detailed information on the *Event Machine* see the *Event Machine* section on the *Configuration* chapter.

¹² Reporting messages can be send to TCP and UDP hosts, to mobile numbers via SMS and to the unit's serial port.

The Extended Reporting Messages (Extended EV)

When using an *Extended Reporting EV Message*, extra tags are added at the end of the standard EV message presented above. These tags allow the addition of information on:

- Inputs and Outputs state.
- Counters' values: Timers and distance counters.
- Vehicle's acceleration.
- Altitude (AMSL).
- Analog to Digital Converter value.
- Back-up battery level.
- Number of GPS-satellites in view.
- GPRS/GSM network state.
- Short Cell ID information.
- Full Cell ID information.
- Vehicle's information based on OBD parameters.
- Virtual Odometer value.
- Region report information.

The information tags are messages separated by the ';' symbol. They are displayed after the *Source* and *Age* fields of the EV message and the last of the tags is the ";ID=" tag which has always been present on every unit's report. One example of such a report is:

```
>REV381447147349+2578250-0802813900000012;AC=000;AL=02595;[...]  
[...]SV=08;ID=EXAMPLE<
```

One extended message can contain one extra tag or it can contain all available tags. The user decides which tags the unit must add to an EV message.

For information on how to make the unit generate *Extended EV Messages* see the *Configuration* chapter and the XAEF and ED messages.

3.11.2 Responses to TAIP Commands Messages

Since Firmware 5.21, the unit allows the report of response generated by a TAIP command included on an event's user-action. This enables the unit to auto-query a TAIP command based on the *Event Machine* and have the response to the command reported to any destination(s). This feature broads the reporting possibilities of the unit as the reporting messages are not only restricted to the information contained on the EV and ET messages. Any information that can only be known by means of a TAIP consult can be converted into a new report message just by adding the message to the

user-action field of an event definition.

For more information and examples see the *Event Machine* section on the *Configuration* chapter.

3.12 Reports' buffer

The *Antares SBTM* reserves a memory space to save reports whenever the remote *Destination* is not available. The buffering process for a *Destination* takes place on any of the following situations:

- For IP-type destinations:
 1. The GPRS session is down.
 2. The GPRS session is up but the unit is on an Over The Air Firmware Upgrade process.
 3. When using TCP: The GPRS session is up but the TCP connection with the IP-host is closed and cannot be established or the host is unreachable.
 4. When using UDP: The GPRS session is up but the IP-host is not acknowledging UDP reports¹³.
- For Telephone Numbers destinations:
 1. The unit is not registered on the GSM network
 2. There is no Telephone Number defined on the report's *Destination*.
- The serial port destination is never buffered.

The unit has an event buffering capacity of 1137 events. Events are buffered and transmitted on an individual manner for each *Destination* meaning that an unreachable destination may have many pending events while other destinations have their buffer empty with no interference between the two destinations. The total buffer capacity whatsoever is shared among all destinations meaning that the maximum holding capacity is always restricted to the free space left by the destination with the maximum accumulated reports.

Note:

The buffering capacity may be significantly lowered by a misconfigured or permanently-unreachable *Destination*.

The buffer is automatically controlled by the unit according to the situations described above and whenever the conditions change for good the unit starts emptying the buffer as messages are successfully transmitted to destinations.

The state for the buffer for a given *Destination* can be consulted with the **XASF** message.

¹³ The UDP acknowledging mechanism is optional and can be turned off/on.

3.13 GPS Back Log

A log with up to to the last minute of GPS information can be retrieved at any time. Combined with acceleration signals to monitor negative accelerations that can occur on possible crash conditions, the Back Log enables a minute by minute examination of the vehicle's location and speed prior to the incident.

The GPS Back Log is retrieved with the **XAKL** message. If no index is given, 60 GPS messages are sent. A minor number can be specified to retrieve less data.

For example, to retrieve the last 20 GPS information messages, send to the unit:

```
>QXAKL0020<
```

To combine this with a large negative acceleration condition, see the *Using Acceleration Signals* section on the *Configuration* chapter. For more information see the **XAKL** message.

3.14 Virtual Odometer

The Virtual Odometer is used to count the total travelled distance. Meaning that its value will be preserved even after an unexpected power loss reset. The Virtual Odometer is configured with the **XAVO** TAIP message. This TAIP message also configures up to 5 thresholds associated with the L signals, which can be used to create reports. The ;VO Extended EV-message Format can be used to consult the value of the Virtual Odometer. For more information on the Virtual Odometer, refer to the **XAVO** TAIP message.

3.15 Authentication Mechanism

An AVL server can use the Authentication Mechanism to validate incoming TCP connections from *Antares SBTM* to prevent fake reports.

Antares SBTM must be configured with a password that is known by the server. The password can not be consulted in *Antares SBTM* by any means. The AVL server will validate the connection by sending a random challenge text to *Antares SBTM*. This challenge text and the password will be used to calculate a MD5 checksum value which *Antares SBTM* will send back to the server. The server will then compare the value received from *Antares SBTM* with its own calculated MD5 checksum value and depending of the result the server can reject the connection or perform any other action configured in the server.

The **XAPW** TAIP message is used to configure the password in *Antares SBTM*. The **XAAU** TAIP message is used to send the challenge text to the server. The **XAMD** TAIP message is used to consult the MD5 checksum value of any string. Please refer to these TAIP messages for more information on the Authentication Mechanism.

3.16 SMS Alias

Antares SBTM allows to configure up to ten SMS Alias. A SMS Alias allows to associate a regular text message with a TAIP message. When *Antares SBTM* receives the associated text message via SMS, it will execute the TAIP message. For information on how to configure the SMS Alias refer to the XATA TAIP message.

3.17 SMS Messages Gateway

The SMS Messages Gateway allows *Antares SBTM* to send any text message up to 80 characters long through a TAIP message. Please refer to the XASG TAIP message for more information on the SMS Messages gateway mechanism.

3.18 Garmin Devices Support

The Garmin devices can be used as communication tools between the driver and the AVL server. The Garmin device allows an *Antares SBTM* user to send and receive text messages from or to a server. It also allows to receive Stop Messages from the server, that can help create routes for the vehicle to follow. *Antares SBTM* can work with any Garmin device that have the Fleet Management Protocol implemented.

The user will be able to send and receive text messages to and from an AVL server. *Antares SBTM* will also send unsolicited status messages each time an action (read, delete, etc) is performed on the text messages stored in Garmin's Inbox.

The AVL server will be able to send and consult the state of Stop Messages. The Stop Messages give the Garmin device coordinates of a destination which it can use to trace a route that the driver can follow. *Antares SBTM* will also send unsolicited status messages when an action is performed on the Stop Message, or when the destination is reached.

3.18.1 Setting up *Antares SBTM*

Antares SBTM communicates with the Garmin device via its serial port at 9600 bps 8n1. *Antares SBTM* needs to be set in the Garmin communication mode using the XAGME message. In this mode the *Antares SBTM* TAIP console will be disabled and it will not accept any TAIP configuration messages via the serial port. The TAIP configuration messages can still be sent over the air to *Antares SBTM*. To exit the Garmin mode via the serial port a special escaping message has to be sent. The escaping message is "EXIT_COMMDATA" and must be sent as a single packet using a communication software like Hyperterminal. It is also possible to disable the Garmin mode over the air with the XAGMD TAIP message.

The Garmin mode and the MDT mode MT are mutually exclusive. The *Antares SBTM* will return Error 91 if the MT message is used to enable the MDT, OBD or Orbcomm mode while the Garmin mode is enabled and viceversa.

3.18. GARMIN DEVICES SUPPORT

The Garmin device must have the Fleet Management Protocol version 1.0 or later to be compatible with the *Antares SBTM* unit. The Fleet Management Protocol is composed of several internal protocols (shown in the table below), which could be implemented or not depending on the model of the Garmin device. The protocols that are necessary to work with *Antares SBTM* implemented functionality are:

| Implemented Functionality | A602 | A603 | A604 |
|---|------|------|------|
| Send text messages to Garmin. | X | | X |
| Receive text messages from Garmin. | X | X | |
| Consult the state of a message previously sent to Garmin (Read, Unread, Deleted). This state is reported either on-request or unsolicited. | | | X |
| Set canned messages for quick replies. | | | X |
| Delete canned messages. | | | X |
| Send Stop Messages to Garmin (Destination Messages) | | X | |
| Consult the state of a Stop Message previously sent to Garmin (Active, Inactive, Done, Deleted.) This state is reported either on-request or unsolicited. | | X | |
| Delete a Stop Message previously sent to Garmin. | | X | |
| Driver ID. | | | X |
| Driver Status. | | | X |
| Canned Replies. | | | X |
| Fleet Managment Protocol data deletion. | | X | |

If a functionality has more than one X mark it means that a Garmin device must have at least one of these protocols so *Antares SBTM* can use the functionality. If the Garmin device has more than one of these protocols, *Antares SBTM* will use the most recent one. Most of the newer Garmin devices have all three internal protocols implemented. For more information about the protocols supported by each Garmin device model, please consult the Garmin device user manual.

The communication between *Antares SBTM* and Garmin is done via a cable with a RS-232 connector and a connector for the Garmin device which varies according to the Garmin device model. The Garmin device is powered with 12V. The cable used for the connection between *Antares SBTM* and the Garmin device has a vehicle-lighter connector to power the Garmin device.

Some of the Garmin devices that have the Fleet Management Protocol and their respective communications cable reference can be found here:

<http://www8.garmin.com/solutions/commercial/supportedproducts.jsp>

<http://www8.garmin.com/solutions/pnd/supportedproducts.jsp>

These are just some of the devices. For more information about devices that support the Fleet Management Protocol please contact Garmin directly.



Part Number: 010-11232-00

nuvi® 205



nuvi® 205W



nuvi® 255W



nuvi® 465T



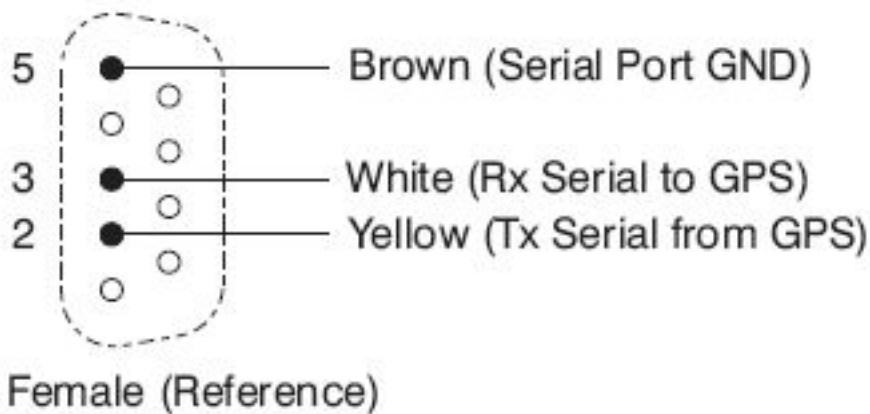
nuvi® 265T



nuvi® 500



Data Cable Pin Out:



3.18.2 Data Flow Example

This is an example of the flow of the data between Garmin, Antares and the server listening for incoming messages from the Garmin device.

To send a text message to the Garmin device from the server, use the following TAIP message:

3.18. GARMIN DEVICES SUPPPORT

>SXAGMTS0000001244751579Hello World<

If the message was received by the Garmin device, Antares will respond to the server:

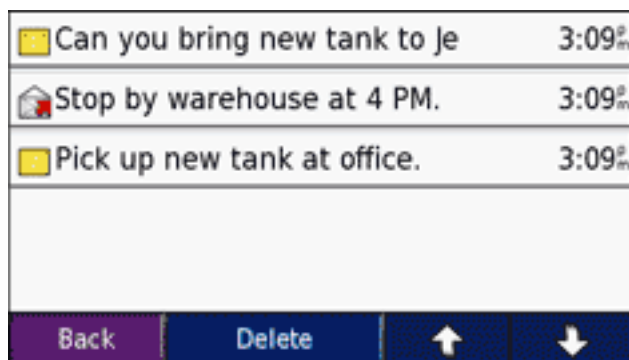
>RXAGMTS0000001244751579Hello World<

If there is a communications problem and the Garmin device does not receive the message, Antares will respond with Error 46 (Internal communications problem).

The Garmin device will display a floating icon indicating that a message has arrived:



When the user presses the icon the inbox will be open showing all the messages present in the Garmin device:



Then the user will touch the new message in the list and the message will be shown:



At the same time Antares will send a status message to the server indicating that the message was read:

```
>RTXGMTS000001<
```

If the user decides to delete the message, Antares will send a message indicating this action:

```
>RTXGMTS000002<
```

To send a message using the Garmin device the user simply needs to type in a message using the Create Message option:



When the user finishes typing the message and touches the Done option, the message will be stored in the outbox and Antares will send the message to the server in a format like this:

```
>RTXGMTR00000001170936526626Hello Server!<
```

The server must be configured to receive unsolicited messages or else the message will be lost. Antares will not send any confirmation message when an action is performed on a message from the Garmin's outbox.

Another option to send messages from the Garmin device is using the Garmin's canned messages:

| |
|--------|
| No |
| May be |
| Later |
| |
| Back |

The user just needs to select the desired message and select Yes when prompted:

| | | |
|---------------|------|-----|
| Send Message? | | |
| "May be" | | |
| No | Edit | Yes |
| Back | | |

Antares will send the message to the server like this:

```
>RTXGMTR00000001180936526626May be<
```

To use the Stop Messages functionality, the server must first send the Stop Message using the XAGMS TAIP message:

```
>SXAGMS0001244753308+0307600189-0957793861127<
```

Antares will send two messages. The first message will be a confirmation that the Stop Message was received:

```
>RXAGMS0001244753308+0307600189-0957793861DCT<
```

If there is a communications problem and the Garmin device does not receive the message, Antares will respond with Error 46 (Internal communications problem). The second message will be an unsolicited message indicating the state of the Stop Message. A new Stop Message is marked as Unread Inactive by the Garmin device:

```
>RTXGMSS000102000<
```

The Garmin device will show a floating icon indicating that a new stop message has arrived:



When the user touches this icon it will show the Stop Messages list on the Garmin device:

| My Stops | | |
|----------|------------------------------------|------------|
| 1 | ✓ Package pickup at Garmin | 63.7° W |
| 2 | ➔ URGENT: Pick up Mr. Lee befor | 7.3° SE |
| 3 | 🚚 Ship with cargo number 57820 | 64.7° N |
| 4 | 🚚 Collect refrigerator: ask for Mr | 64.7° N |
| Back | | Delete All |

Then when the user selects the new Stop Message from the Stop Message list, Antares will send an unsolicited message to the server indicating this action:

```
>RTXGMSS000103000<
```

If the user chooses to follow this Stop destination, Antares will confirm that the stop message was set as active with the an unsolicited message:

```
>RTXGMSS000100000<
```

Or if the user deletes the message, Antares will send the unsolicited message to confirm this action:

```
>RTXGMSS000104---<
```

Please refer to the XAGM TAIP message for more information on the configuration of the Garmin device with *Antares SBTM*.

3.18.3 Setting Up The Server

The server that will send and receive the messages from and to *Antares SBTM* when using the Garmin device must be configured to receive unexpected messages (asynchronous communication). Meaning that it must receive and processes correctly the status messages and text messages sent at anytime by *Antares SBTM*.

3.19 ORBCOMM Satellite Modems Support

Use Orbcomm System satellite modems as a backup to send event reports when a GPRS-IP destination is unreachable.

The destination of the reports when using the Orbcomm System satellite modem must be configured in both the satellite modem and Orbcomm's server by the user because *Antares SBTM* does not configure the destination points used by the satellite modem. The satellite modem serial port must be configured to work at 9600bps.

Depending on the configuration script, *Antares SBTM* can send the TAIP EV message exclusively to the satellite modem through the serial port when there is no communication between the AVL server or other IP destination points. The TAIP EV message may contain the vehicle information such as Speed, Acceleration, Position, Ignition state, etc.

In this basic integration, *Antares SBTM* will not monitor the state of the Orbcomm system. *Antares SBTM* will send the message through the serial port to the modem and will wait for confirmation that the message was received by the modem before deleting it from its buffer. Because of this it is important that the modem is correctly configured and has been tested to be working correctly by prior to connecting it to *Antares SBTM*.

The event definition must be done using the Message ID qualifier "O" from the ED TAIP message, which will enable *Antares SBTM* to send an EV reporting message to the serial port using the SC-Originated Default Message protocol from the Orbcomm Serial Interface Specification. This message can be up to 116 bytes long. The destination address of the message must include the serial port. This can be done by using the Event Handling "X" which will send the message to the serial port or a destination address that contains the destination point P15 (serial port). When using a destination address that contains both an GPRS-IP destination and the serial port, *Antares SBTM* will send a SC-Originated Default Message to the serial port and a regular EV TAIP message to the GPRS-IP destination.

3.19.1 Setting up *Antares SBTM* for ORBCOMM Satellite Modems

Antares SBTM communicates with the satellite modem via its serial port at 9600bps 8n1. To do so, *Antares SBTM* serial port must be configured to work on an special byte mode which disables the TAIP console. The default state of *Antares SBTM* is to work the serial port on TAIP console mode, not in byte mode. Byte mode has to be enabled with a TAIP command so that *Antares SBTM* can communicate with the Orbcomm System satellite

modem. To enable the byte mode for Orbcomm System satellite modem use the >SMTR< TAIP message.

When in byte mode the user can not configure or consult the unit locally (The TAIP console is always available over the air). To enable TAIP console over the serial port again, an special escaping message has to be sent. This message will disable byte mode and enable TAIP console.

The escape message to enable the TAIP console is “EXIT_COMMDATA” without the quotes and in uppercase. This message has to be sent in a single packet over the serial port. This means that writing EXIT_COMMDATA on Hyperterminal by hand won’t serve as an escape message. To send the message as a single packet, write EXIT_COMMDATA on any text editor, select and copy the text, then go to Hyperteminal and after checking the correct port, baud rate (9600) and connecting to the COM port, select Edit, Paste to Host. If the escape sequence is received correctly, *Antares SBTM* will respond with the message “EXIT_OK”. Then, TAIP commands can be exchanged with the unit.

Once enabled, the byte mode will be retained on resets and power-cycles

3.19.2 Operation

Antares SBTM will send a TAIP EV messages to the Orbcomm system satellite modem through it’s serial port using a serial cross over cable. Only the Tx, Rx and GND lines are needed for communication.

Antares SBTM encapsulates a TAIP EV message for the satellite modem using the SC-Originated Default Message defined on Orbcomm’s Serial Interface Specification, when an event that has been defined with the Message ID qualifier “O” is triggered.

More information about the SC-Originated Default Message from Orbcomm’s Serial Interface Specification can be requested directly to Orbcomm at Service.customer@orbcomm.com. However this information is not required to set up *Antares SBTM* to work with the satellite modem.

If the Event Handling field is set as “X” the encapsulated message will only be sent to the serial port. If the event is defined using an Destination Address that contains both an GPRS-IP destination point and the destination point P15 (Serial port), the unit will send the SC-Originated Default Message encapsulated EV message to the serial port and a regular EV message to the IP address.

3.19.3 Example

On the following example we will create a tracking report that depends on two time periods. One time period for when the vehicle is traveling and another time period for when the vehicle is not moving. The period of time for when the vehicle is traveling will depend on the connectivity to an GPRS-IP destination. To determine whether the vehicle is traveling or not, we will set a Start Condition and a Stop Condition using a low speed limit that assumes that the vehicle is not moving. However this script will not report each time the Start and Stop condition are met. It only uses

this conditions to change the report criteria intelligently. In this example the report event will be sent to the serial port only when the GPRS-IP destination is unreachable.

First we must create the IP destination point.

```
>SXADP0000avl.server.com;1234<
```

And a destination address with the destination point.

```
>SDA0;P00<
```

Then we must create a Start/Stop condition. First we define a speed limit of 5mph to determine if the vehicle is traveling or not traveling.

```
>SGS0510050<
```

When the vehicle falls below this speed, we will start a counter of 60 seconds to prevent that the Start/Stop Conditions are triggered by quick speed changes.

```
>SED10SV0;S05-;ACT=SGC05TC00060<
```

When the 60 second counter is complete the Stop condition is met and we will set User Signal 05 to false which we will use later on.

```
>SED11SV0;C05+;ACT=SSSU050<
```

The Start Condition will be met when the vehicle goes over the speed limit after a Stop Condition. We will then set User Signal 05 to true.

```
>SED12SV0;S05C05&;ACT=SSSU051<
```

Now that we have our Start/Stop Conditions set, we must create a condition that determines whether the IP destination is reachable and when it is unreachable. To do so, we'll use the A00 signal. When Signal A00 transitions to false it means that the connection to the destination point 00 (our AVL server in this case) was lost (only true on TCP). To prevent that too many messages are sent to the satellite modem due to quick socket connection losses we will use a 10 minute counter that will remain true while the A00 is false to determine that the IP destination is in fact unreachable. The A00 will also transition to false if the unit is having problems with GPRS.

```
>SED13SV0;A00-;ACT=SGC00TC0001000060<
```

When the 10 minute timer is completed we will set User Signal 00 to false.

```
>SED14SV0;C00+;ACT=SSSU000<
```

When Signal A00 transitions to true it mean that the connection to the IP destination has been reestablished. We will delete the 10 minute counter and set User Signal to true.

```
>SED15SV0;A00+;ACT=SGC00U;ACT=SSSU001<
```

In this example we will use Counter 01 for the periodic reports. We will use

3.20. OBD SUPPORT

a 60 minute timer for when the vehicle is not traveling. We will use this time whether the GPRS-IP destination is reachable or unreachable.

>SED16SV0;U05-;ACT=SGC01TR0006000060<

Then we will set a timer for when the vehicle is traveling and the IP destination is reachable. We will use a 5 minute timer for this example.

>SED17SV0;U05U00&+;ACT=SGC01TR0000500060<

And a event for when the vehicle is traveling but the IP destination is unreachable. We will use a 20 minute timer to prevent that too many messages are sent using the satellite modem.

>SED18SV0;U05U00!&+;ACT=SGC01TR0002000060<

We need to delete the timer that determines the Stop Condition when the vehicle exceeds the speed limit. We do this using a greater Event ID than the one used to determine the Start Condition to prevent that the Start Condition is not met since it depends on the C05 Signal.

>SED19SV0;S05+;ACT=SGC05U<

Define the events that will send the reports. One for when the IP destination is unreachable that will only send the message to the satellite modem using the SC-Originated Default Message protocol. Notice that we use the “O” qualifier for the Message ID field.

>SED49X00;U00!C01&+<

And one that will send a regular EV TAIP message to the IP destination when it is reachable.

>SED00NV0;U00C01&+<

Antares SBTM must be working on Byte Mode so it can communicate with the satellite modem.

>SMTR<

3.20 OBD Support

Through the *OBDII Interface model BG* accessory, *Antares SBTM* can use the vehicle’s On-Board Diagnostic (OBD) system parameters to generate reports based on the vehicle’s Engine RPM, Throttle Position, Speed, Acceleration, Odometer, Fuel Level, Fuel Rate, Ignition State and Malfunction Indicator Light.

Not all parameters are supported by all vehicles and not all vehicles are supported. This depends on the brand/model of the vehicle and also on the OBDII Interface model.

For more information about the OBD support, please refer to application note “AN0014E OBD Interface BG for *Antares SBTM*” which can be downloaded [here](#).

3.21. FIRMWARE UPGRADE

Also refer to the XAOS, XAOE, XAOG, XAOF, XAOR, XAOT TAIP messages which are used to configure the *Antares SBTM* signals associated with the OBD parameters and the MT TAIP message which enables the communication between *Antares SBTM* and the *OBDII Interface model BG* accessory.

3.21 Firmware Upgrade

The *Antares SBTM* firmware may be changed locally over the serial port or remotely with the unit's built-in *Over The Air Firmware Upgrade* support.

3.21.1 Over The Air

Depending on the selected APN's Internet access the unit may be commanded to initiate an over-the-air firmware upgrade process using *Digital Communications TechnologiesTM*'s servers or if no Internet access is available for the units the user may create a firmware server for its units to use as upgrading servers.

Note:

For the last method contact *Digital Communications TechnologiesTM* for instructions and support.

Warning:

Not all of the units have the over-the-air firmware upgrade feature enabled. TAIP error 69 or 90 is returned when trying to upgrade one of these units. The feature can be enabled with instructions from DCT.

The first method is preferred as it only requires the use of a single command for the unit to begin and manage the whole upgrading process.

This process is started with the XAFU message. See the TAIP reference for detailed information. The message takes a firmware version number for example 5.22, a flag that tells which server to use: DCT's or any defined by the user, and a *Destination Address* or *Destination Point* to send the upgrade progress state.

A normal update process showing diagnostic messages over the serial port is something like:

```
>SXAFU00522SV15;ID=0000<
>RXAFU00522SV15;ID=0000<
>RTXFW update: Starting.;ID=0000<
>RTXFW update: Connected.;ID=0000<
>RTXFW update: Download started.;ID=0000<
>RTXFW download progress: (235495/20368)B, 8%, 2546B/sec;ID=0000<
>RTXFW download progress: (235495/40736)B, 17%, 4073B/sec;ID=0000<
>RTXFW download progress: (235495/61104)B, 25%, 5092B/sec;ID=0000<
>RTXFW download progress: (235495/81416)B, 34%, 4062B/sec;ID=0000<
```

3.21. FIRMWARE UPGRADE

```
>RTXFW download progress: (235495/101840)B, 43%, 4084B/sec;ID=0000<
>RTXFW download progress: (235495/122208)B, 51%, 4073B/sec;ID=0000<
>RTXFW download progress: (235495/142520)B, 60%, 4062B/sec;ID=0000<
>RTXFW download progress: (235495/162944)B, 69%, 4084B/sec;ID=0000<
>RTXFW download progress: (235495/183256)B, 77%, 4062B/sec;ID=0000<
>RTXFW download progress: (235495/203680)B, 86%, 5106B/sec;ID=0000<
>RTXFW download progress: (235495/224048)B, 95%, 4073B/sec;ID=0000<
>RTXFW update: Download OK.;ID=0000<
>RTXFW update: Installing (module will restart).;ID=0000<
>RTXFW update: New firmware installed!. Running version:
                Antares GPS 05.22;ID=0000<
```

How does it work After entering the XAFU command the update process is queued until the unit has GPRS access. When GPRS is available it starts connecting to the firmware server. Then the unit will try to connect and download the selected firmware a finite number of times.

The unit downloads the new firmware in a safe mode: In a separate memory place to guarantee that if the download process gets interrupted there will be no loss of the unit's functionality.

The unit will inform the end of the process to the *Destinations* selected with the XAFU command.

Remarks

- After the file is downloaded the unit automatically installs and runs the new firmware.
- The download process may take 2 or 4 minutes under good network conditions.
- After downloading and installing the new firmware, the unit will restart but no buffer data (pending events) will be lost.
- The unit's configuration is preserved throughout firmware upgrades¹⁴.

Example Initiate an over-the-air firmware update with DCT's servers in order to upgrade the unit's firmware to version 5.22 having the diagnostic messages sent over the serial port:

```
>SXAFU00522SV15<
```

If no previous upgrade process is taken place the unit returns:

```
>RXAFU00522SV15<
```

Indicating that the command was accepted and it has queued the process until GPRS is available (if not available yet).

At a later time the unit will start showing diagnostic messages on the form of TX messages. For example:

¹⁴ There may be some exceptions to this. Consult *Digital Communications TechnologiesTM* for information

3.22. TAIP DOWNLOADERTM TOOL (WRITE/READ SCRIPTS)

```
>RTXNew firmware installed!. Running version: Antares GPS 5.22  
<
```

Meaning that the upgrade process ended ok.

For detailed information see the XAFU message.

3.21.2 Upgrading locally

This upgrade is done through the unit's serial port, with a terminal software like WindowsTM, HyperterminalTM. A .hez file containing the firmware is required.

Note:

For the firmware file and instructions contact *Digital Communications TechnologiesTM*.

3.22 TAIP DownloaderTM Tool (Write/Read scripts)

Use this tool to Write or Read a configuration script to/from an *Antares SBTM*. A configuration script is a plain-text file that holds TAIP configuration messages on each line with the possibility of including user comments that are not passed to the unit on a Write process.

The software uses a command-answer mechanism and a retry-timeout mechanism that is suitable for communicating with the unit's TAIP console.

For more information on creating, reading and writing scripts refer to the *Using Scripts* section of the *Configuration* chapter.

The TAIP DownloaderTM software can be downloaded for free from:
<http://www.digitalcomtech.com>

Warning:

Make sure of using version 1.0.7 or superior of *TAIP DownloaderTM*. Contact *Digital Communications TechnologiesTM* for information on how to upgrade.

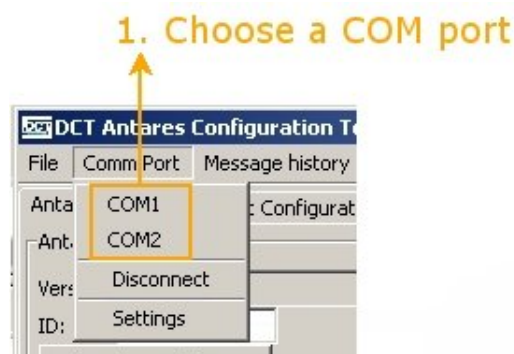
To install *TAIP DownloaderTM* simply run the installer. The tool can be opened from Windows, Start, All Programs, TAIP Downloader.

3.22. TAIP DOWNLOADERTM TOOL (WRITE/READ SCRIPTS)

3.22.1 Communicating locally with the *Antares SBTM*

Power on the unit. Connect a direct serial cable between the unit's and the PC's serial ports. You can also use an USB serial port converter¹⁵. Open *TAIP DownloaderTM* and follow the next steps.

3.22.2 STEP 1. Selecting a COM port



The *Comm Port* menu shows a list of serial ports detected by *TAIP DownloaderTM* at startup. Before selecting a serial port, make sure that other applications like Hyperterminal are not using it. If your serial port is not listed, close *TAIP DownloaderTM*, check your serial port hardware and open *TAIP DownloaderTM* again. When the port is successfully opened, a black dot is displayed next to its name.

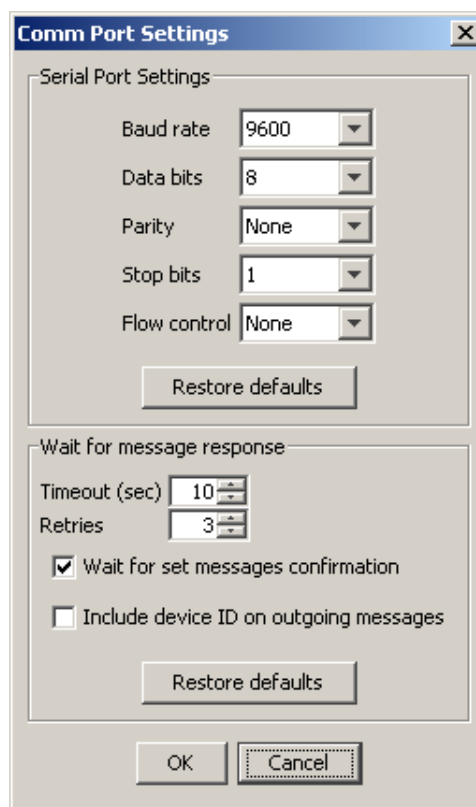
3.22.3 STEP 2. Test Communication

Try to communicate with the unit by clicking "Test Communication". The unit's Version, ID and IMEI fields should be filled. This test may fail if:

- The selected COM port is not the port the unit is connected to.
- A virtual USB COM port may need to be reset: Close the application, disconnect the USB/Serial Converter, wait a few seconds, connect again, wait for Windows to detect the hardware and reopen *TAIP DownloaderTM*. If this fails, try disconnecting the USB/Serial converter and reinstalling its drivers.
- The unit is temporarily unavailable to attend the TAIP console: Retry the test after 15 seconds.
- *TAIP DownloaderTM* default configuration has been changed: Check that "Comm Port", "Settings" are set like the next diagram:

¹⁵ *TAIP Downloader'sTM* list of available COM ports is only updated at start-up. If you create/connect a new virtual (like USB) port, it won't be listed until a software restart.

3.22. TAIP DOWNLOADERTM TOOL (WRITE/READ SCRIPTS)



- Finally, contact DCT.

At this point you have successfully communicated with the unit and any configuration or reading process can take place.

3.22.4 Write a Configuration Script

You can now configure *Antares SBTM* by Writing a configuration script to the unit. You can create a configuration script from scratch or copy one of several scripts shown on this manual or you can ask for a script file from DCT. For more information on script refer to the *Using Scripts* section on the *Configuration* chapter.

To pass a configuration script to the unit, on *TAIP DownloaderTM* select "Device", "Write configuration". *TAIP DownloaderTM* will ask for an script file. This is generally file having the ".tmf" extension but it could be any plain-text file with a set of TAIP configuration messages. After clicking "Open", the writing process starts. Depending on the script size this could take from 5 to 40 seconds. The communication process is shown on the "TAIP message history" field. If a TAIP command defined on the script is not recognized by the unit, *TAIP DownloaderTM* alerts the user and asks him whether to "Cancel" the writing process or to skip the message causing the error.

3.22. TAIP DOWNLOADERTM TOOL (WRITE/READ SCRIPTS)

3.22.5 Read a Configuration Script

When reading an *Antares SBTM*, an script file (tmf file) containing an unit's configuration is created. This script can be edited and passed to other units to replicate a configuration. To generate an script file based on an unit's configuration, on *TAIP DownloaderTM* select "'Device'", "'Read configuration'". This process takes approximately 1 minute.

3.22.6 Over The Air

TAIP DownloaderTM can not communicate remotely with devices by itself. But using third party softwares, a virtual serial port can be created, so that the virtual communication is done via TCP or UDP. This enables reading and writing scripts Over The Air.

One of such tools that creates a virtual serial port connected trough a TCP connection is HW Virtual Serial Port which can be found at:

<http://www.HW-group.com>

Remember that the *Antares SBTM* works as a TCP client so HW Virtual Serial Port has to be configured as server. To do so make sure to select the option HW VSP works as the TP Server only box on the software.

For more information on these software tools contact *Digital Communications TechnologiesTM*.

4 Configuration

The *Antares SBTM* configuration is done through the unit's TAIP console via TAIP commands. The TAIP console can be used with several communication methods:

- The unit's serial port.
- TCP or UDP (GPRS level) communication.
- 2-way SMS (GSM level).

This means that the unit is configured *over-the-air* in the same way as it is done locally.

As described in the *Operation* chapter the TAIP console works on a *question-answer* basis. So when configuring the unit a similar *configuration-confirmation* schema has to be followed. This means that a configuration message which is really a TAIP *set message*¹ has to be responded by the unit with exactly the same *response message*² before it can be assumed that the unit accepted the configuration message. If this confirmation fails a retry-timeout mechanism has to be followed.

All of the configuration messages are immediately saved on persistent memory (they will not be lost when the power is lost) and with a few exceptions described on the TAIP reference all commands take effect immediately.

As long as the *configuration-confirmation* mechanism is implemented it is irrelevant if the configuration commands are sent manually or if a script file (tmf file) is sent with the *TAIP DownloaderTM*³ software.

The order of the configuration messages is also irrelevant but the user needs to take care of any inconsistency due to an “incomplete” configuration process that leads to a parameter depending on an un-configured value.

Not all of the unit's parameters are required for it to start working. The full spectrum of configuration messages is covered on the *Unit's TAIP reference* chapter. Almost all of the unit's behavior is controlled by the configuration given on the *event machine*, but there are some other parameters needed to get the *Event Machine* properly working. These parameters enable the unit on the *GSM/GPRS* network and define *IP hosts* and/or *phone numbers* to be used by the *Event Machine* as *destinations* for its reports. *Destinations*

¹ A TAIP *set message* is characterized by the letter S on the message's qualifier. Refer to the *Operation* chapter for more information.

² A TAIP *response message* is characterized by the letter R on the message's qualifier. Refer to the *Operation* chapter.

³ *TAIP DownloaderTM* Tool. Refer to the *Operation* chapter

4.1. *UNIT'S ID

are also defined in order to gain remote access to the unit.

Note:

The sections on this chapter marked with an * are essential to every configuration.

Note:

This chapter is a complement of the *TAIP reference* chapter and vice versa. A lot of times will be expected for the reader to continue the given explanation on the TAIP reference when making reference to a TAIP message/command.

4.1 *Unit's ID

(*)*Essential configuration* This parameter is only meaningful to the AVL software which is going to receive reporting messages from the unit. It is not necessary for the unit to work but it may necessary for making a *Getting Started* test with an AVL software.

The unit's ID is a 10 characters maximum string containing any character but ';', '<' or '>'. The initial value is 0000.

The ID is used every time the unit sends a TAIP report message (EV or ET) by adding the postfix ";ID=UNIT'S ID" to the message. This postfix gives the AVL software information on who is sending the report.

The AVL software may use the >QID< query to ask for the unit's ID once and only a new TCP connection is established. The ";ID=" postfix may be then eliminated from the unit's reports with the RM message, this saves a few consumption bytes.

Refer to the TAIP reference for more information. The ID setting takes the following form for an unit's ID UNIT-0015:

>SIDUNIT-0015<

4.2 *Enabling the unit on GSM and GPRS

(*)*Essential configuration* The GSM registration as well as the GPRS attachment (getting a valid IP address from the cellular operator) is automatically done by the unit all the time. This means that whenever there is GSM and GPRS network availability and the unit is correctly configured it is going to be GPRS-attached (The green On Line LED *solid* or *blinking*) and GSM-registered (The orange signal LED *blinking* or *off*). For this, two parameters are

required: The SIM card's user PIN and the cellular operator APN (Access Point Name).

4.2.1 SIM Card's PIN for GSM registration

This parameter is *optional* because it depends on how the SIM card is configured. Most cellular operators distribute no-pin SIM cards so you don't have to worry about this parameter or you can explicitly set it to "empty" if you are unsure of a previous configuration⁴. But if your SIM card requires a PIN this parameter has to be set or else the unit will fail to register on GSM. The configuration message for the SIM card's PIN is the **RF** message. You can check the TAIP reference for more information but the command's use is as follows:

To set the PIN number 1234 send to the unit:

```
>SRFI1234<
```

To set an "empty" PIN:

```
>SRFI<
```

Use the **Q** qualifier to consult.

A PIN configuration can be issued at any time but although the registration process is always done automatically, the unit will take up some time on registering to the GSM network when a previous erroneous PIN was given or when no PIN was not given. So it is recommended in those cases to reset the unit after the PIN-set command with the **>SRT<** reset message. Or better, have the PIN correctly configured before the SIM card is inserted.

Note:

The *Antares SBTM* does not block a SIM card when the given PIN is wrong.

The GSM registration status can be consulted with the **RP** message and/or with the *Signal* led. For more information consult the TAIP reference and the *Leds* subsection on the *Operation* chapter.

When the unit is GSM-registered it is able to make or receive telephone calls and 2-way SMS communication.

Note:

The PIN parameter can not be changed over the air. This to prevent the unit from losing the GSM/GPRS network when an incorrect PIN is tried remotely. The PIN value can only be modified over the serial port.

⁴ Any PIN value will work when using a no-PIN SIM card

4.2.2 Access Point Name (APN) for GPRS set up

A GPRS session enables the unit to communicate with IP networks (IP addresses). Although the most common IP network is the Internet, by specifying an APN the cellular carrier can enable your unit(s) to work on a user-private IP network that does not *necessarily* communicate with the Internet. Among other advantages discussed on the *Operation* chapter, when using a private APN, devices like the Antares, Cellular Phones, PDAs, PCs, Servers, etc can only be accessed by devices on the same APN. There are several reasons why a cellular carrier may use different APNs, but the important thing is that ***an APN is needed for the unit to start a GPRS session***, for it to be able to send and receive data over IP networks. The most common situation is an APN with Internet access. Any device using this APN has the ability of communicating with any IP network on the Internet. Specifically if you are running your AVL (Automated Vehicle Location) server on the Internet, this is the kind of APN you want.

An APN has the form of a server name on a dot-separated format and it is supplied by the cellular carrier.

For example:

```
this.is.an.apn.com  
or  
internet.carrier-name.com
```

Setting the APN on the *Antares SBTM* is also done with the **RF** message. An “empty” APN may be configured too. An empty value is used when a GPRS session⁵ is not desired. For more information refer to the TAIP reference, but the command takes the following form:

To set the APN to *internet.carrier-name.com* send to the unit:

```
>SRFAinternet.carrier-name.com<
```

To leave the APN parameter empty:

```
>SRFA<
```

You can check the GPRS registration status with the **XANS** message and/or with the *On line* led. This led has to be either ON or blinking when GPRS is up. For more information consult the TAIP reference and the *Leds* subsection on the *Operation* chapter.

An APN configuration can be issued at any time and the unit will start registering to the GPRS network as soon as the GSM registration process is done and the APN parameter is set.

Note:

The *Antares SBTM* will not start a GPRS session if it is not registered on the GSM network. And when the GSM network is lost the GPRS session is lost too. However the unit may work on the GSM network regardless of the GPRS session state.

⁵ GSM communication only: Voice and/or SMSs

Warning:

The GPRS registration process is usually charged by the cellular carrier, having an incorrect APN will make the unit constantly try on failing GPRS sessions which could lead to an excess on the unit's bytes consumption.

Note:

The APN value can not be changed over the air when communicating via IP networks. This to prevent losing communication with the unit over GPRS when a wrong APN is used. The APN can be modified over the air by means of SMS communication.

Once the GPRS session is up the unit is ready to communicate with IP networks (i.e. with IP addresses). For this, the cellular carrier assigns the unit an IP address that is usually but not necessarily dynamic, meaning that for every session the unit starts its value changes. The actual IP address assigned by the operator through the chosen APN can be consulted with the XAIP command.

4.3 *Destinations (DPs and DAs)

(*)*Essential configuration* A destination makes reference to the device or user that is receiving and/or sending messages from/to the *Antares SBTM*. The possible destination for the unit are:

- 10 IP hosts. Via TCP and/or UDP.
- 5 telephone numbers via SMS.
- The unit's serial port.

Each destination is called a *Destination Point* (or simply a DP). *Destination Points* may be grouped to form a *Destination Address* (or simply a DA). As you can deduce from the list, there are 16 DPs. The unit offers 10 possible combinations of DPs, leading to 10 DAs. In the majority of cases DAs are used to tell the unit where to send its report but sometimes (i.e. some commands need) an specific destination (a DP) is required.

4.3.1 Destination Points (DPs)

As mentioned there are 16 DPs, these are divided in the following way:

- The first 10 (DPs 00 to 09) are IP hosts. These are defined with

4.3. *DESTINATIONS (DPS AND DAS)

an IP address or a server name⁶ and a TCP or UDP port number. The *Antares SBTM* is a TCP and/or UDP⁷ client which always starts the communication. This means that the IP host has to be a TCP or UDP server listening for incoming connections on the same port specified here.

- The next 5 (DPs 10 to 14) make reference to phone numbers. These numbers are used to send SMSs or make voice calls. They are also used as authorization numbers for responding to received SMSs commands and/or answering incoming voice calls. Defining if a report should be sent on TAIP or using a custom user message is also done here.
- The last DP (15) makes reference to the serial port.

For more information see the XADP message.

4.3.2 Destination Addresses (DAs)

A *Destination Address* is a user-defined group of *Destination Points*. This enables some reporting commands to route their report to several destinations at the same time with a single definition. Up to 10 (0 to 9) DAs may be defined. Refer to the DA message for more information. This command enables the user to group a list of DPs and/or a range of DPs.

The main use for DAs is on the *routing options* of an event definition. The *Event Machine* section gives more information about this. What should be clear on this, is that *a report generated by an event is always sent to a DA, not to a single DP*. For this reason DAs make part of the minimum configuration required by the unit. Some examples of DAs' definitions are:

1. Defining DA 5 as the group containing DPs 04, 06, 10 and 15:
>SDA5;P04,P06,P10,P15<

This will make any event using DA 5 as *Destination Address* on its routing options to send the same report to the IP host 04, IP host 06, phone number 10 and the unit's serial port. Such an event could be defined as:

>SED23NV5;TD1+<

2. Defining DA 8 as the group containing DPs 00 to 03, 07 to 09 and 14:
>SDA8;P00:P03,P07:P09,P14<

3. To delete a DA definition:
>SDA8U<

⁶ Make sure of using the eXtended version of the DP message when working with names.

⁷ The unit can be set to listen for UDP queries on an UDP-server manner. See the XAUP and XAUN messages.

Note:

You can always define a DA containing a single DP so you can send a single report to a single destination. For example: >SDA3;P01<

For more information see the DA message.

4.4 Reporting

The reporting criteria depends solely on the *Event Machine* configuration and on the parameters that controls how signals used by the events' triggers change. Events' *routing options* allow the generation of reports to several destinations including IP addresses, phone numbers (thru SMSs) and the unit's serial port for accessories like PDAs. To understand how reporting configuration is done, refer to the next section called *Event Machine* that teaches on how to tell the unit the *what*, *where* and *when* of the user-defined reports.

The reporting messages generated by the *Event Machine* to the *Destinations* are the TAIP messages EV and ET. These messages contain among other GPS-related information the event code generating the report. For information on these messages refer to the TAIP reference.

For SMS' destinations the EV and ET TAIP messages are used when an AVL application has access to a SMS server system⁸ capable of receiving the message and analyze it in the same way as does when it comes from an IP channel. When the SMS destination is not an AVL server but a person's phone number, a user-defined text message associated to an event code can be configured to be sent instead of the not-user-friendly TAIP message. To do so the XATM message is used and the *Destination Point* configuration has to be altered to tell the unit to use user messages on a given DP instead of TAIP reporting messages. See the XADP message for more information on this.

Note:

The *Antares SBTM* reporting messages that have to be interpreted by the AVL software server are the EV and ET messages. See the TAIP reference when developing an AVL application capable of extracting the report information from these messages.

See also the *Reports' messages* section on the *Operation* chapter.

⁸ Using Short Message Peer-to-Peer Protocol (SMPP) or a dedicated GSM modem connected via USB or serial port.

4.5 *Event Machine

(*)*Essential configuration* The unit's reporting is controlled by an *Event Machine* which constantly evaluates user defined events. These events allow the user to create a reporting schema and functionality controlled by *triggers* and *actions*. Events can be consulted or configured at any time with the ED message throughout the TAIP console, enabling the user to alter the *Event Machine* parameters at any time *locally* or *over the air*.

Up to 50 events may be defined on the unit. These events are evaluated on a sequential order based on the event's ID. This means that lower IDs are evaluated first. Having this in mind an event's ID may be relevant if its trigger depends on other events' signals and/or on other events' user-defined actions. The whole 50 events' evaluation is performed at a 1 second rate.

This section examines the events' components: *triggers* and *actions*. Then it gives an overview of the events' definition to complement the TAIP reference. Finally explains one of the most important components of the events' triggers: ***Signals***.

Note:

The *Event Machine*, based on user's configuration decides what, where and when to send a report and/or execute an action, ***but it is not all of the configuration needed by the unit***. When the user starts configuring the *Event Machine*, it is assumed that he has already defined the *Destination Addresses* and their corresponding *Destination Points* as described on the previous sections of this chapter. Enabling the unit on the GSM/GPRS network is also required.

4.5.1 Triggers

A trigger is determined with the *logical combination* of several situations (also called signals). A *logical combination* is basically an equation (specifically: a boolean equation) that combines signals (situations) with the logical operators AND, OR and NOT. In Antares, these boolean equations use the post-fixed notation, meaning that the operator is at the end of the signals to be evaluated. When more than three signals are being evaluated, a logical operator must be inserted every two signals in the equation. These are some examples of the postfix notation syntax:

A or B $\rightarrow AB||$
A and B $\rightarrow AB\&$
A and B and C $\rightarrow AB\&C\&$

To determine how the signals will trigger the report a plus (+) or minus (-) sign is added at the end of the equation. A plus sign (+) indicates that the report is generated when a signal or an equation becomes "true". Con-

sequently, a minus (-) sign indicates that the report is generated when the signal or the equation becomes “false”.

When A **or** B becomes true $\rightarrow AB||+$

When A **and** B **and** C becomes false $\rightarrow AB\&C\&-$

If the report must be generated when one signal becomes true and another becomes false one of the signals must be negated using the boolean operator **not**. Either the plus or minus sign can be used, but for it is easier to understand the equation when the plus sign is used.

When A becomes “true” and B becomes “false” $\rightarrow A!B\&+$

Example Combining situations with operators:

A vehicle going at or over 60mph is a situation, let’s call it situation **S**. This situation may be *true* or may be *false*, there are no other possibilities. These kind of situations that can only take two possible values, *true* or *false*, are said to be of *boolean* nature. Let’s add another **boolean** situation: A vehicle located within some cities’ perimeter. This again is a two-value-only situation. Let’s call it **P**.

How can we combine these two situations to create a third complex situation ?. We can do so with logical operators. For example, let’s create a third situation **C** that is only true when the other two situations are also true. For this we use the **AND** operator in the following form:

C = S AND P

Now, every time **S** and **P** are both true at the same time, **C** is going to be true too. Whenever **S** or **P** become false, **C** will become false.

The situation we just created (named **C**) is useful for detecting an speed limit violation within a cities’ perimeter. With **C** we specify that we are interesting not only on detecting a general speed violation but an specific speed limit for an specific city boundary. Note that a cities’ perimeter is just an example: We could have specified a road or a small neighborhood.

You can create simple triggers which go off whenever a single specific situation occurs, like when a vehicle’s emergency button wired to a unit’s input is activated. Or you can construct more complex triggers by combining situations with the logical operators named above like it was done on the previous example. Having this is mind different triggers may be created to accomplish tasks like:

- Sense a *panic button* to send the current vehicle’s position to different destinations, including the vehicle’s owner cellular phone (through a SMS).

Have the unit make a voice call is also possible. This will make the cabin’s audio available to a predefined phone number whenever the driver presses the alarm button.

- Speed limit monitoring which includes a time condition (seconds) and a visual advisory (using an output) that gives the driver a chance to slow down before the actual speed violation is reported.
- Generate an alarm report whenever the primary power is disconnected. The optional unit's internal back up battery enables the unit to keep on working even when the vehicle's battery is disconnected. This prevents thieves to easily disable power to the unit.
- Generate a report when a vehicle goes out of a predefined polygonal region.
- Sense and report the back-up battery level status.
- Use different reporting criteria according to the time of the day.
- Generate a reconnection message to a TCP server whenever the socket gets reconnected.
- Generate a distance report whenever the vehicle's traveled distance reaches a predefined threshold. (The virtual gps-based odometer is not 100% precise)
- Use the communication channels available to the unit on a cost effective way as they start failing: You can enable the unit to always send its report via GPRS and only use SMSs when GPRS goes down, finally if all GSM goes down too you can attach another type of last option communication media like a satellital modem to the unit's serial port and tell the unit only to use it when both GPRS and GSM are down. This gives a full communication coverage at a cost effective schema.
- Sense the vehicle's ignition and speed to create IDLE, STOP and MOVING events.
- Reduce a report frequency when the unit starts roaming on GSM.
- Sense the primary power presence and go to low power consumption mode or on the contrary, increase the reporting criteria to report it as an abnormal condition.
- Create an intelligent *Time and Distance* report which automatically increases the reporting criteria to a top as the vehicle travels fast and decreases it to a minimum when the vehicle travels at low or zero speeds.
- Create a confirmation report that confirms that a remotely-set output has reached the set state.

4.5.2 Actions

Once you have defined a trigger (and/or a set of triggers) the next step for configuring the *Antares SBTM* event machine, is to tell the unit what to do when a trigger goes off. There are two types of actions the unit may take when a trigger goes off. These are the *report* action and the *user-defined* action.

Report action

When the *event machine* detects that a trigger goes off it uses the configuration of the *report* action to generate a report. A *report* action configuration includes *routing options* and a *type of report*.

The *routing options* tell the unit where to report the occurrence of the specific event. The report destinations may be⁹ IP addresses (or host names), cellular phone numbers, unit's serial port. It could also be a silent report which is not reported to any destination¹⁰.

The *type of report* is used to tell the unit which kind of TAIP report message generate when reporting to a destination. The unit supports two reporting messages:

- ET Time-only report. Use T as Message ID on an *event definition*. See the ED message.
- EV Event report: Use V as Message ID on an *event definition*. See the ED message.
- extended-EV: As mentioned on the *Reports' messages* section of the *Operation* chapter, the EV report can include extra information tags. To do so, the Message IDs A, B and C of an event definition are used. The information tags for messages IDs A, B and C are defined with the XAEF command. For more information see the ED, EV and XAEF TAIP messages. See also the *Scenarios and examples* section.

User-defined action(s)

An user-defined action is defined by a user-specified TAIP message (or set of messages) that are appended at the end of a regular event definition using the "ACT=" string. This enables the user to predefine TAIP messages¹¹ that the unit will only process whenever the trigger goes off. This is specially useful when making the unit drive outputs automatically. For example: The unit may automatically (without asking a remote server/user) shut a vehicle's engine off¹² whenever a vehicle is trying to move on a forbidden out-of-labor time. It is also helpful when driving the unit's *counters* to create complex reporting scenarios.

Any TAIP command available on the TAIP console can be predefined as

⁹ With the *routing options* a single destination or a group of destinations may be specified. This makes the unit able of sending the same report to several destinations at the same time.

¹⁰ This feature is useful when using events to trigger other events or when a *user-specified* action is specified but no reporting is desired

¹¹ Several TAIP actions can be set on an event definition. See the ED message for information.

¹² **Shutting an engine off without considering a vehicle's state is not advised.** See the *Scenarios'* chapter for better ways of doing this.

an user action, so there is a wide range of possibilities to choose. Some of them include:

- Change a reporting schedule according to a special region and/or time of day.
- Start a situation timer: For example to keep track of the time a security door remains open. Another example of using timers and events is to keep track of a vehicle's IDLE situation like when a vehicle remains at rest with the ignition ON and/or monitor a STOP situation meaning the vehicle is at rest and the ignition is OFF.
- Keep track of how many times a situation occurs. Like keeping track of how many times a speed limit is violated.
- Lock or unlock the vehicle's doors.
- Go to sleep mode to preserve the unit's back-up battery.
- Start a voice call to a predefined phone number.
- Activate a led connected to any output so the driver knows when there is a voice call being held to any of the authorized phone numbers.
- Activate a led connected to any output whenever a vehicle is out of a polygonal geo-fence.
- Have the *Antares SBTM* safely¹³ turn the vehicle's engine off whenever this travels too far from the normal working region.
- Enable a vehicle's ignition when the working hour begins.
- Set/Reset user signals to trigger other events. This allows the creation of complex event's scenarios.
- Sense the vehicle's ignition to have the unit start working on a low power consumption mode whenever the vehicle is off.

User-actions (pre-set TAIP commands) can also be used to generate reports. This is because the response to a TAIP command(s) that is defined on an event's user-action field is (are) sent to the event's *Destination Address* (for non-silent events). So, any information that is not included in the reporting messages ET, EV and extended-EV, that can be consulted with a TAIP query, can be reported by creating an event that has the TAIP query or queries on the user-action field.

For example, the following configuration >SGC00TR00300<

>SED00NV0;C00+;ACT=QRP;ACT=QGC06V<

will make the unit report the GSM network state (>QRP<) and the value of the counter 06 >QGC06V< every 5 minutes to *Destination Address 0*.

The user-actions can also be silent, meaning that unlike the previous example, the actions will not generate a report. This silent actions allow to

¹³ The user configuration should include a low speed precondition. **Turning an engine off without knowing a vehicle's state is not advised.**

4.5. *EVENT MACHINE

perform actions that do not need to be reported. To define a silent action use the “XCT=” string. For example, the following configuration:

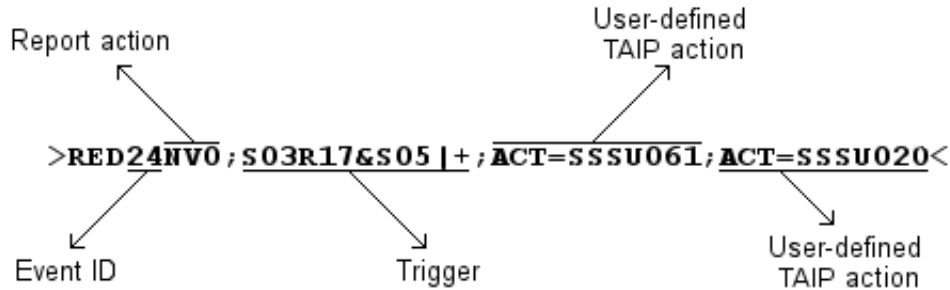
```
>SED01NV0;F00+;XCT=SGC01TR00300<
```

will make the unit set a 5 minute counter everytime the vehicle is turned on, but there will be no response for this action . Only the vehicle ignition state will be reported.

For more examples see the *Scenarios and examples* chapter.

4.5.3 Events

Finally, *triggers* and *actions* are bound together on a single configuration message called *event*. An event is defined or consulted with the TAIP ED message. A single event holds a *trigger*, a *report action* and optionally an *user action*. The next figure gives a global description of the TAIP ED message. For detailed information refer to the unit’s TAIP specification chapter.



The *Antares SBTM* has 50 events available for the user to configure. They may be defined all at once in a *configuration script* or they may be individually defined at any moment as the user adds/removes functionality. The actual events’ definitions of the unit may be consulted with the TAIP message `>QED<`. This will have the unit deliver the configuration state of all 50 events. An example of the returned configuration on the TAIP console is:

```
>QED<
```

```
>RED00NV0;A00TD0&F00&+<
>RED01NV1;A00!F03&TD1&F00&+<
>RED02XM0;F03!TD2&F00&+<
>RED03NV2;G00+<
>RED04NV0;A00TD3&F00!&+<
>RED05NV1;A00!F03&TD3&F00!&+<
>RED06XM0;F03!TD3&F00!&+<
>RED07NV0;A00U00&+;ACT=SSSU000<
>RED08NV1;A00!F03&U00&+;ACT=SSSU000<
```

```
>RED09U<
>RED10NV0;A00U01&+;ACT=SSSU010<
>RED11NV1;A00!F03&U01&+;ACT=SSSU010<
>RED12U<
>RED13U<
>RED14U<
>RED15U<
>RED16U<
>RED17U<
>RED18U<
>RED19U<
>RED20NV0;A00C02&+<
>RED21NV1;A00!C02&+<
>RED22XM0;F03!C02&+<
>RED23U<
>RED24U<
>RED25U<
>RED26U<
>RED27U<
>RED28U<
>RED29U<
>RED30U<
>RED31U<
>RED32U<
>RED33U<
>RED34U<
>RED35U<
>RED36sV0;S00-;ACT=SGC02U<
>RED37sV0;S00+;ACT=SGC02TC00010<
>RED38U<
>RED39U<
>RED40sV0;F00+;ACT=SSSU001<
>RED41sV0;F00-;ACT=SSSU011<
>RED42U<
>RED43U<
>RED44U<
>RED45U<
>RED46U<
>RED47U<
>RED48U<
>RED49U<
```

You can see some events having a user-defined TAIP action, different routing options and many undefined events (having a “U” (for undefined) after the event ID). For more information on how to interpret this reading as well as how to create such configuration refer to the ED message.

4.5.4 Signals

As described in the previous section, the *event machine* takes actions like *reporting* or *switching outputs* whenever a user defined trigger goes off. This trigger is configured by the user with the logical combination of situations.

4.5. *EVENT MACHINE

A situations makes reference to a vehicle state which is in fact represented by **signals** and their state. *Antares* SB^{TM} signals are of boolean nature, meaning that they can only take one of two possible values: *true* or *false*.

Signals and the logical operators **AND**, **OR**, **NOT** are used to create logical equations to form events' triggers.

By using the **SS TAIP** message a signal's state can be consulted, and depending on the signal's type, this command can be used also to change the signal's state.

Note:

| |
|--|
| <i>Signals</i> ' names always have three characters. |
|--|

The available signals for the *Antares* SB^{TM} are presented in the next table.

4.5. *EVENT MACHINE

| ID | Signal | Description |
|-----------|--------------------------------------|---|
| A00 - A09 | <i>Destination Points'</i> state | True when the IP address/port defined on the corresponding <i>Destination Point's</i> index is accepting a TCP connection or when using UDP (i.e. the TCP/UDP socket is open). See the XADP message. |
| B00 - B04 | Battery Levels(+) | True when the unit's back-up battery level is above the value defined with the XAGB message. |
| C00 - C19 | Counters, Timers, Distancers(+) | True when the corresponding counter reaches its defined threshold value. See the GC message. |
| D00 - D04 | ADC Levels(+) | True when the ADC input voltage level is above the value defined with the XAGA message. |
| E00 - E49 | Event Triggers(+) | True when the corresponding event trigger is <i>True</i> . See the ED message. |
| F00 | Ignition | True when the ignition input of the unit is on. |
| F01 | GPS Fix | True when doing GPS fixes. |
| F02 | GSM/GPRS Roaming | True when the unit is Roaming on GSM/GPRS. |
| F03 | GSM-Registered | True when the unit is registered in the GSM network. |
| F04 | GPS Antenna Feed-line fault | Indicates a short on the GPS antenna cable. |
| F05 | GPRS-Registered | True when the unit is <i>registered</i> on the GPRS network. |
| F08 | GPRS-Attached | True when the unit is <i>attached</i> to the GPRS network. |
| F09 | MDT PAD Mode | True when the unit is in MDT PAD mode. See the MT message |
| F10 | PAD message | True when an MDT message is received on the serial port when PAD mode is ON. See the MT message |
| F11 | Woke-Up | True when the unit wakes-up after sleep power mode. Immediately reset after the first events' evaluation. See the XAPM message |
| F12 | 12volts/24volts detector | True when the main supply voltage is above 16 volts. |
| F13 | Power | True when the unit's main power supply is on. |
| F14 | Cell ID change | This signal transitions to True when the unit registers in a new cell. |
| G00 - G07 | General Purpose Input Outputs(*)(-). | These signals are true when the corresponding GPIO is true. See the SS and GF messages. |
| H00 - H04 | Store & Forward Thresholds | True when the amount of messages stored in the S&F Buffer exceeds the corresponding threshold value. See the XAGF message. |
| IP1 - IP4 | Inputs. | True when the corresponding input is on. See the SS message. |
| J00 - J04 | Heading Deltas. | True when the vehicle's heading change is greater than the corresponding heading change threshold. The signal is immediately reset after evaluation to achieve a turn-by-turn report. See the SXAGH message. |
| K00 - K99 | Circular Regions. | True when the vehicle is inside the corresponding Circular Region. See the SXAGR and XAIR messages. |
| L00 - L04 | Virtual Odometer Thresholds | True when the virtual odometer value exceeds the corresponding threshold value. See the XAVO message. |
| N00 - N04 | Acceleration. | For positive acceleration thresholds: True when the vehicle's acceleration is larger than the corresponding threshold. For negative acceleration thresholds: True when the vehicle's acceleration is less than the corresponding threshold. See the XAGN message. |

4.5. *EVENT MACHINE

| ID | Signal | Description |
|-----------|--------------------------------------|---|
| OE0 - OE4 | Engine's RPM thresholds | True when the engine's RPM detected by the OBD is higher than the corresponding RPM thresholds. See the XA0E message. |
| OT0 - OT4 | Throttle Position thresholds | True when the throttle position percentage is higher than the corresponding throttle thresholds. See the XA0T message. |
| OF0 - OF4 | Fuel Level percentage | True when the remaining fuel in the vehicle drops below the corresponding fuel thresholds. Percentage. See the XA0F message. |
| OG0 - OG4 | Fuel Gallons thresholds | True when the remaining fuel in the vehicle drops below the corresponding fuel thresholds. Gallons. See the XA0G message. |
| OR0 - OR4 | Fuel Rate thresholds | True when the fuel consumption is higher than the corresponding fuel rate thresholds. Gallons per hour. See the XA0R message. |
| OS0 | Status: Communicating with Vehicle | True when the OBD is communicating with Antares. See the XA0S message. |
| OS1 | Status: Communication Checksum error | True when an OBD communication checksum error has occurred. See the XA0S message. |
| OS2 | OBD Vehicle Ignition State | True when the OBD detects that the vehicle ignition is on. See the XA0S message. |
| OS3 | Vehicle Malfunction Indicator Light | True when the OBD detects that the Vehicle Malfunction Indicator Light (MIL)(Check Engine Indicator) is on. See the XA0S message. |
| OS4 | Secondary OBD tool connected | True if there is a secondary OBD tool connected. See the XA0S message. |
| R00 - R29 | Regions(+) | True when the vehicle is inside the corresponding region. See the GR message. |
| RST | Software reset | This signal is used to detect a firmware reset. It is set to true every time there is a software reset and it is immediately reset after the first events' evaluation. |
| S00 - S09 | Speed thresholds(+) | True when the vehicle's speed is faster than the corresponding speed threshold. See the GS message. |
| T00 - T09 | Time Windows(+) | True when the time and date are within the corresponding time window. See the GT message. |
| TD0 - TD9 | Time and Distance(+) | True when the associated <i>Time and Distance</i> counter has a Time and Distance condition true. The signal is immediately reset after being evaluated to enable the counter for further triggers. See the TD message. |
| U00 - U09 | User flags(-) | These signals may be changed by the user at any time with the SS message. |
| V00 - V05 | Voice signals | V00 signal is <i>true</i> when there is a voice call on course. V01 to V05 are true whenever the actual voice call is being held to <i>Destination Point</i> 10 to 14 respectively. See the VC, VE and XADP messages. |
| XP1 - XP4 | Outputs(-) | True when the corresponding output is on. See the SS message. |

(*) *Antares SBTM* does not have GPIOs. These signals are maintained for backwards compatibility with scripts from other versions. Instead, *Antares SBTM* has 4 inputs and 4 outputs. Signals G00-G03 are the same as IP1-IP4 signals and G04-G07 signals are the same as XP1-XP4 signals. The user is encouraged to use IP and XP signals instead of G signals.

(-) These signals' state can be directly altered by the user with the SS TAIP message.

(+) Refer to the next paragraph.

The signals marked with a (+) are signals for which the conditions that set them *true* or *false* are defined by the user. The other signals transitions' criteria can not be altered as they depend on conditions already programmed on the unit. Next a list of the (+) signals with their corresponding configuration message:

- Counters' signals with the GC message: The corresponding signal is *true* whenever the counter value exceeds the threshold configured with the GC message.
- Events' signals with the ED message: These signals' state depend on the logical state of the associated event trigger.
- Regions' signals with the GR message: The signal is *true* whenever the vehicle is inside the region defined with the GR command.
- Speeds' signals with the GS message: The signal is *true* whenever the vehicle's speed exceeds the threshold configured with the GS message.
- Time Windows' signals with the GT message: The signal is *true* whenever the actual date/time lies between the minimum and maximum date/time defined with the GT message.
- Time and Distance' signals with the TD message: The signal is *true* when the Time And Distance condition configured with the TD message is met. This signal has the particularity that it is immediately set to *false* after a complete evaluation of all the events. This creates a "periodic" sort of signal which is suitable for a Time And Distance report.

An example of using these messages to define a signal:

Example Defining the speed signal S06 to be *true* whenever the vehicle's speed exceeds 55mph and false whenever the speed is below:

```
>SGS0610550<
```

Now the speed signal S06 can be used on an event's definition to generate a report every time the vehicle exceeds 55mph:

```
>SED33NT3;S06+<
```

We just defined event 33 to send an ET report to DA 3 every time the S06 signal switches from *false* to *true*, that's what the + sign does. The report will only be generated once. In order for the event to trigger again the S06 signal has to become *false* first and then *true* again. This means that the speed excess is going to be reported every time the speed goes from anything less than 55mph to a larger value but not every time the speed remains at the larger value.

4.5.5 Examples

Some examples about the use of the event machine are presented next. For complete examples containing both the *Event Machine* and the other parameters necessary to get the unit working refer to the *Scenarios and Examples* chapter. To understand the events' configuration format refer to the ED message on the TAIP reference.

Example Configuring two events on the *Event Machine* to generate an ignition report:

The ignition ON event may be defined as:

```
>SED18NV4;F00+<
```

And the Ignition OFF event:

```
>SED19NV4;F00-<
```

Both events' *routing actions* indicate that the destination of the report is the DA 4 and that EV is the reporting message to generate. Both events use a simple trigger consisting of a one-signal-only condition, F00 which is the vehicle's ignition signal.

Example Creating an event to detect when the vehicle leaves a defined region:

When a vehicle is outside the perimeter defined by the polygon that defines a region, the corresponding region's signal is set to false. To generate an event whenever a vehicle leaves the region the region's signal has to be monitored for the *true* to *false* transition (this is done with the minus sign on the event's trigger):

```
>SED24NV7;R26-<
```

This event is going to be reported to DA 7 which has to be defined before the event occurs so the *Event Machine* can generate the report to a valid group of DPs.

The region has to be defined too. To do so, use the GR message. This is the only signal-defining message that requires more than a single command to configure the signal. This because as specified on the TAIP reference, the GR message defines not a whole region but a region's single point. One example for the definition of region 26 using three points is:

```
>SGR26001-8021500+025795009999<
```

```
>SGR26011-8024444+025840550099<
```

```
>SGR2602U-8017861+025849440001<
```

4.6 Using *Polygonal Regions*

The *Antares SBTM* supports detecting when a vehicle is inside or outside of up to 30 user-defined polygons (50 points maximum each). These polygon-defined regions are defined with the GR message and the associated regions' signals are the Rxx signals.

4.7. USING CIRCULAR REGIONS (GEO-FENCES)

Note: For circular regions see the next section and the **XAGR** message

For instructions on creating a region refer to the **GR** TAIP message.

Polygonal regions are useful for creating large regions, like cities' boundaries. A large region can be used to:

- Restrict a vehicle's operation to an specific region by generating an exception report when traveling outside the boundary.
- Restrict a vehicle's operation to an specific region by SAFELY¹⁴ shutting the engine OFF when leaving an special area. The vehicle's operation can be restores later by a time condition or a remote activation.
- Use different tracking criteria according to city/rural areas. For example: A turn-by-turn is useful on the city, whether a Time/Distance condition may suit better on rural areas.
- Use different speed limit violation reporting according to special areas/states.

For check-points, route following and geo-fences, Circular Regions may lead to better results. See the next section.

4.7 Using *Circular Regions* (geo-fences)

100 Circular Regions defined each by a Latitude/Longitude center and a radius in meters can be set. As polygonal regions are useful for creating different criteria depending on different large areas, circular regions are useful for setting check-points for route control, on-time arrival control and geo-fence alarms to restrict a vehicle's movement. Circular regions are set with the **XAGR** message. The triggering signals are the **Kxx** signals where **xx** is the region's index.

For example, to set a Circular Region centered at Lat: 25.782066, Long: -80.281380 having a radius of 200 meters, send to the unit:

```
>SXAGR551+257820-08028130002000000000<
```

Now, to generate event 05 whenever the unit enters the region:

```
>SED05XV0;K55+<
```

See the **XAGR** message for more information.

Creating a region centered in the actual vehicle's position is also supported. The **XAIR** message allows to set Circular Region centered on the actual position with the given radius.

For example, to instruct the unit to create the Circular Region 44 centered "here" having a radius of 500 meters, send to the unit:

¹⁴ Create a shut engine off condition that only acts when vehicle is at rest.

4.8. USING REGION ID REPORTS

>SXAIR44000500<

Note that any region previously defined on the index 44 will be overwritten.

Now, to generate event 30 when the vehicle leaves this region, send to the unit:

>SED30XV0;K44-<

To undefine the region, use:

>SXAGR44U<

For more information see the **XAGR** and **XAIR** messages.

4.8 Using Region ID Reports

Use the Region ID report to define a single event for several circular and polygonal regions. A report will be sent using only one event code when the associated regions' signals transition. The Region ID reports are configured using the **XARE** TAIP message. This TAIP message will generate a report every time the unit enters a region and/or the unit leaves a region. The generated report will include: In/Out indicator, the type of region, and the region index.

For example, to define the Region ID report using the event code 00 that will report both when the unit goes In or Out from a region, use the following configuration message:

>SXAREB00NV0<

A report generated when the unit entered the circular region 10 would be:

>REV001524377378+0475230-0740249100000012;RE=IK10;ID=test<

A report generated when the unit left the polygonal region 20 would be:

>REV001524377378+0475230-0740249100000012;RE=OR20;ID=test<

For more information on the Region ID reports, refer to the **XARE** TAIP message.

4.9 Setting *Speed Limits*

Speed limits are created with the **GS** TAIP message. When creating an speed limit we are telling the unit to activate the corresponding speed signal **Sxx** whenever the vehicle's speed is greater than the defined limit and to get set the signal low whenever the vehicle's speed is below the limit. Up to 10 speed limits may be defined, so there are 10 speed signals (**S00** - **S09**) available to be used on an event's definition.

Example To create the speed limit 00 having a value of 55mph send to the unit:

>SGS0010550<

Now an event can be created so any time the vehicle exceeds 55mph event 33 is generated:

>SED33NV0;S00+<

A configuration may be created so a warning to the driver is given via a LED indicator before actually generating the event. See the *Scenarios and Examples* section for this.

4.10 The *Time And Distance* criteria

Up to 10 *Time And Distance* (T&D) criterion can be defined. *Time And Distance* definitions are covered on the TD TAIP message.

A T&D signal attached to an event generates a T&D-like report whose frequency changes according to the vehicle's displacement. With this, the unit increases the report frequency when the vehicle is moving (accumulating traveled distance) and decreases it when the vehicle is not moving. This is more efficient than having the unit report by a *time-only* criteria were almost all of the reports triggered when the vehicle is not moving are not relevant.

Note:

The unit can also be configured with a *time-only* criteria.

How does this criteria work ?

First, the report has to be triggered by a distance threshold that tells the unit to send a report whenever the accumulated distance exceeds this value. This *distance-only* scheme has two problems:

1. What if the vehicle goes too fast and/or the distance threshold is low ?. It will report too frequent.
2. What if the vehicle moves too slow and/or the distance threshold is large ?. Or worse, what if it does not move at all?. It will report very few times or may never report.

To solve this problem the *distance-only* report is improved with a ***Time and Distance*** report that adds two controlling parameters for the *Distance threshold*. These parameters are the *Minimum time between reports* and the *Maximum time between reports*.

The *Maximum time between reports* takes control of the report when the unit has not exceeded the *Distance threshold* for a long time. So this value ensures a minimum reporting frequency. One way to see this parameter, is that this time will be the report period when the vehicle is stationary.

The *Minimum time between reports* takes control of the report when the unit exceeds the *Distance threshold*. Assuring that even when the unit is exceeding the *Distance threshold* it will not report too frequently, no more frequent than the configured minimum time value. One way of seeing this parameter is that this time will be the report period when the vehicle is moving too fast.

A *Time and Distance* signal is configured with the following parameters:

4.11. USING TIME WINDOWS

1. Distance Threshold.
2. Minimum time between reports.
3. Maximum time between reports.

A T&D signal (TD) gets high according to its defined T&D parameters and it is immediately reset after all events have been evaluated in order to get ready for the activation of the signal again, when the T&D criteria is met on the future.

Example To create a tracking event that sends event code 49 based on a T&D criterion. A 8km (5miles)report is desired. The criterion must be so that no more than one hour passes between successive reports, and the minimum time between successive reports must be 2 minutes.

To do so we choose for no particular reason the T&D criterion 6. The requirements call for the following T&D configuration:

- Minimum time between reports: 120 seconds (2mins).
- Distance Threshold: 80 x 100m (8km).
- Maximum time between reports: 3600 seconds (1hr).

This configuration is achieved by sending:

```
>STD60120000000803600<
```

Now we create event 49 using the TD6 signal:

```
>SED49NV0;TD6+<
```

The T&D could also be configured on a time-only basis by defining the *Minimum Time* parameter and setting the others to zero. To define a time-only criteria of 15 minutes (900 seconds):

```
>STD60900000000000000000<
```

or:

```
>STD60900<
```

4.11 Using *Time Windows*

Refer to the GT message.

4.12 Using *Counters*

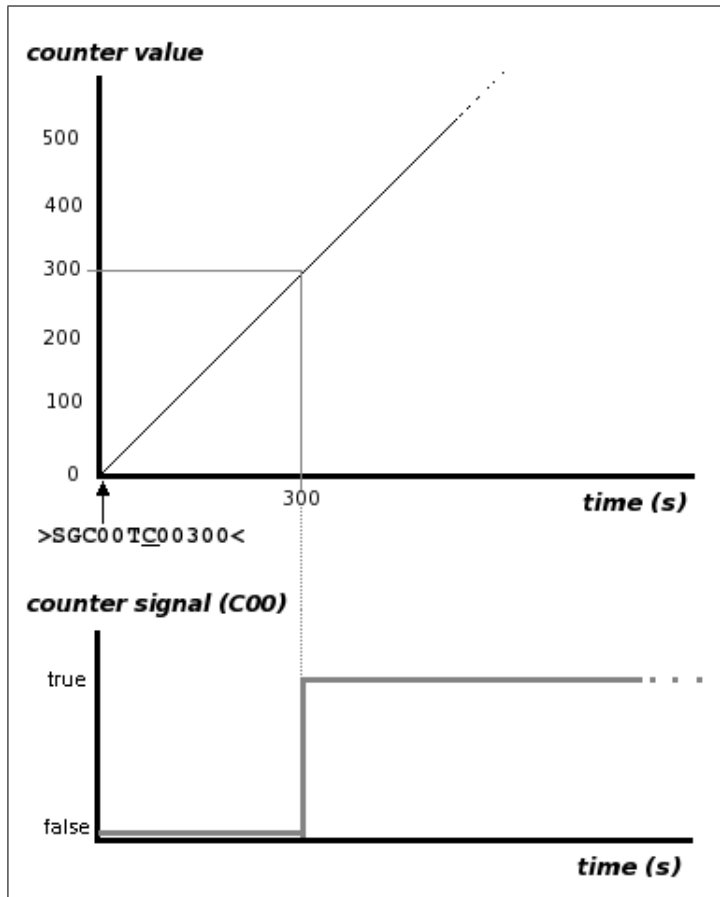
Counters' configuration/manipulation is controlled by the GC message. Up to 20 counters can be configured. For detailed information and more examples see the GC message. Next, a brief description of counters' operation.

There are three types of counters:

4.12. USING COUNTERS

- For counting time (TIMERS).
- For counting distance (DISTANCERS).
- For counting situations (COUNTERS).

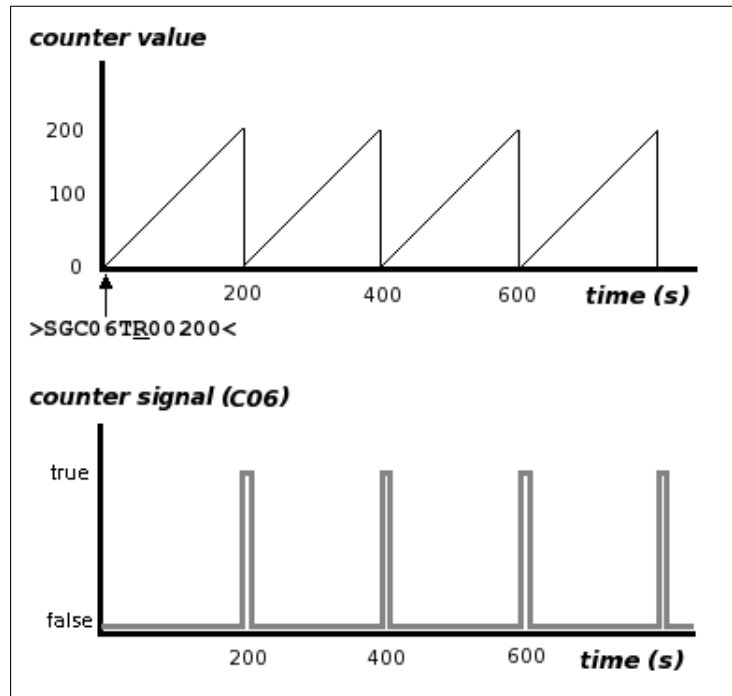
For each counter a counter signal **Cxx** exists. A counter signal gets true whenever the counter's value exceeds a user-defined threshold and gets low on the contrary. For example, a *time* counter (TIMER) set to count seconds with a threshold value of 300 will make a corresponding **Cxx** signal to become true 300 seconds after defining the counter. See the next graphic to illustrate this.



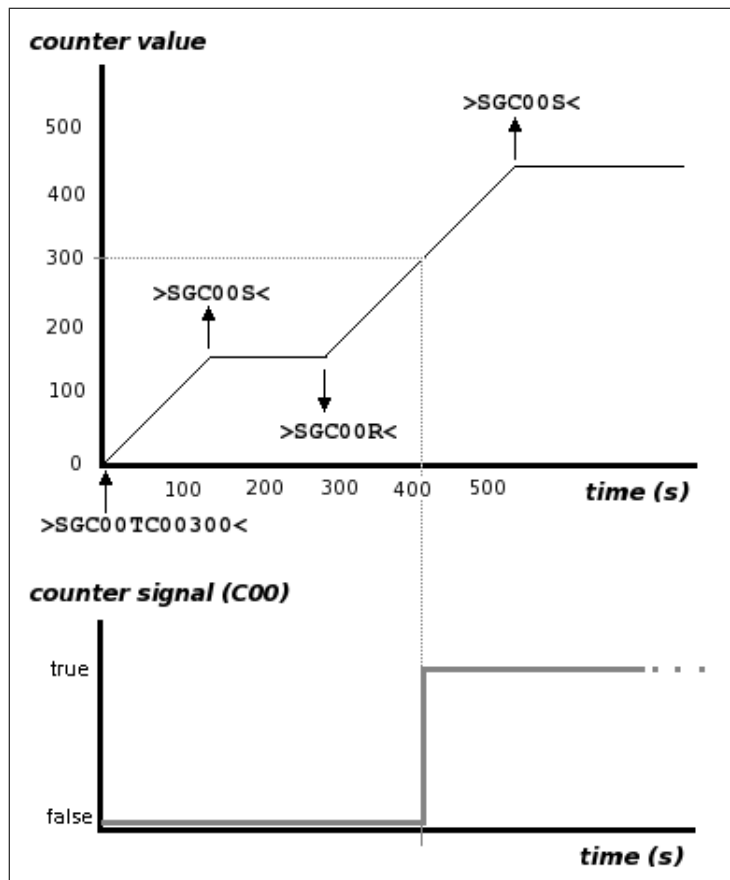
When a counter reaches the defined threshold it can:

- Keep on counting (continuous counter). To define a continuous counter: *Set the recycle flag to C.* The previous illustration shows a *continuous* counter.
- Reset the counter value to zero (recycling counter), creating thus a periodic counter signal. To define a recycling counter: *Set the recycle flag to R.* See the next illustration of a *recycling* counter.

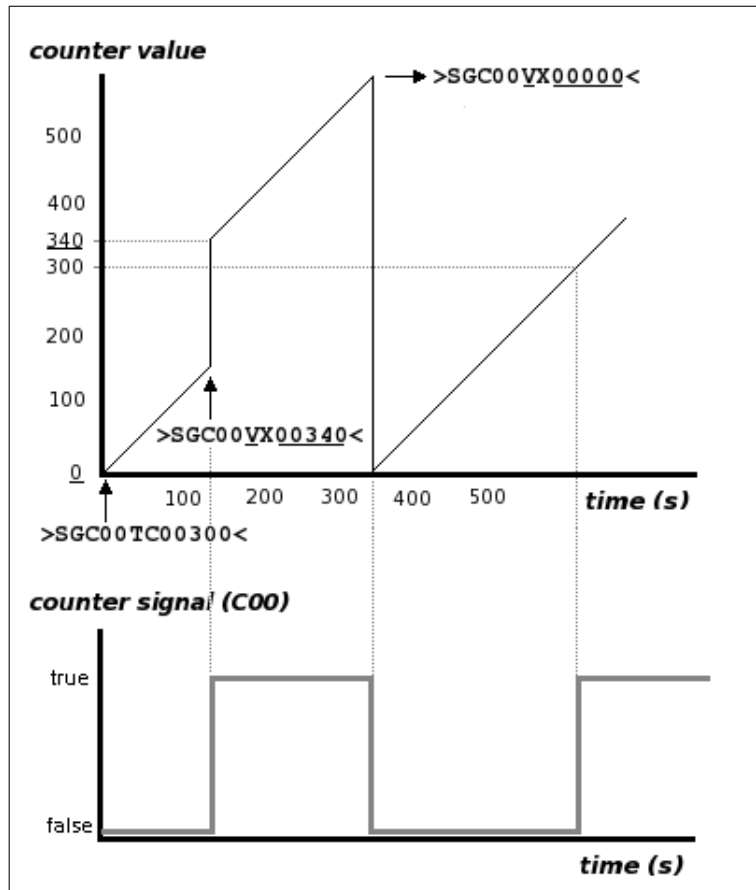
4.12. USING COUNTERS



A counter may be suspended and resumed with the `GC` command. Use the *S* and *R* commands on a previously defined counter. The next graph illustrates this.



A counter value can also be "manually" altered with the **GC** command. Use the **V** command on a previously defined counter. The next graph illustrates this. Note that altering a counter's value does not affect its counting configuration: After altering its value, the counter keeps on counting from the given value.



The X next to the V on the setting-value command indicates the counter to preserve its recycling flag: If the letters R or C were used instead of X, the counter will not only change its value but its recycling configuration too.

A counter value may be consulted at any time. Use the V command in query form (`>QGC00V<`).

A counter configuration may include an optional *delta* parameter to make the counter value increment only when the counting variable reaches the *delta*. This is useful for counting not only seconds but minutes, hours, etc or for distance counter to count meters, kilometers, tens of kilometers, etc.

COUNTERS-type counters only count on a command basis, meaning that count when told to do so. Use the I command.

As with any TAIP command, counters manipulation/configuration can be included on an event's action, so counters can be used to start timers or distance counts on an automatic basis when a defined situation occurs. An example of such configuration can be found on the *Scenarios and Examples* sections. With the same method, counters can be used to count the number of occurrences of an event.

For more examples and information on counters see the GC command on the

Unit's TAIP reference chapter and the Scenarios and Examples sections.

4.13 Manipulating *signals*

Signals marked with a (-) sign on the signals' list on the *Event Machine* section can be directly manipulated by the user. These signals are:

- Outputs (4).
- User Signals (10).

Signals' manipulation and consulting is achieved with the **SS** TAIP message.

When setting/resetting an output with the **SS** message a direct hardware activation/deactivation takes place on the selected output. See the *Outputs* section on the *Configuration* chapter for more information.

4.14 User *signals*

User signals (U) are used to create complex configurations. User signals are set or reset only with the **SS** message. Thus, they are used to trigger events based on a user set/reset **SS** command or on a set/reset **SS** event user-action. Refer to the *Scenarios and Examples* section for more information. The *Safe engine turn-off* example is configured so that a user signal activation starts the *safe engine turn-off* process. This user signal activation is not done on the script as it is not part of the configuration of the unit but of its operation: Whenever a remote system tries to turn-off the engine, a user signal activation is performed by sending the TAIP **SS** command over the air.

4.15 Using *Heading Deltas* (turn-by-turn report)

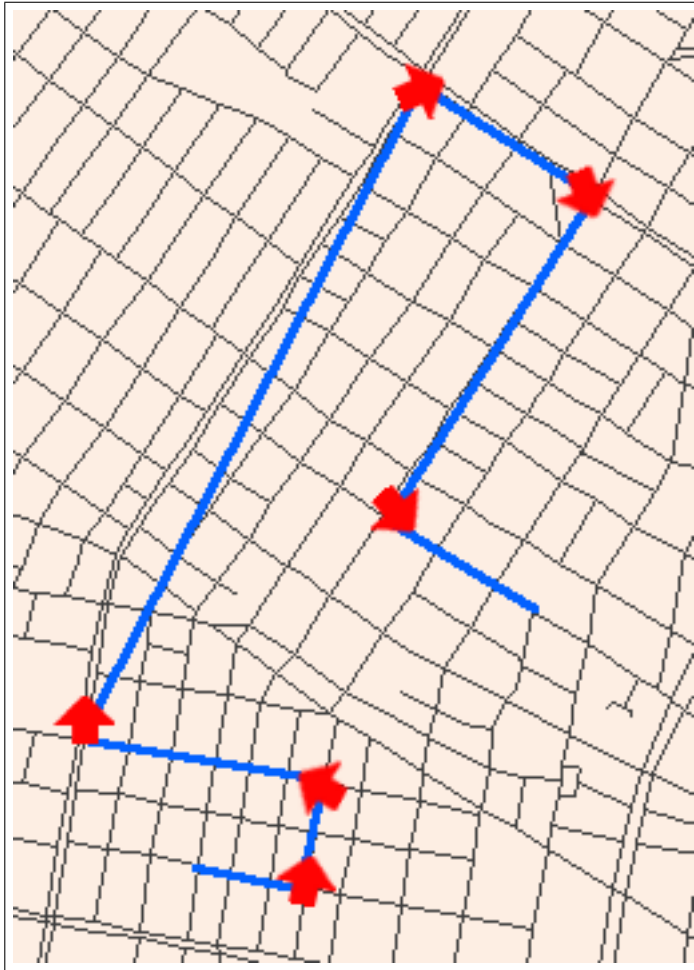
By using *Heading Deltas*, a turn-by-turn reporting criteria can be created. This is achieved by making the unit report only when the vehicle's heading changes significantly since the previous heading-change report. To do so, the *J* signals are used. A *J* signal is automatically set high whenever the vehicle's heading changes by more than the delta value defined with a **XAGH** message. After being evaluated, the signal is set to low to enable further triggers. Also, the heading value used for comparing the actual heading of the vehicle is updated so a new heading change can be detected.

Note:

Heading-change detection is only done when the vehicle's speed is greater than 5mph.

A turn-by-turn report, having a delta value of 45 degrees, looks like:

4.16. DRIVING METRICS (ACCELERATION, MAX. SPEED, ETC)



Up to 5 heading deltas may be defined. The minimum value for a heading delta is 15 degrees and the maximum is 180. A typical value that suits a turn-by-turn report is 45 degrees. Here is an example of creating a turn-by-turn report using 45 degrees as heading change to trigger event 40:

Set a heading delta of 45 degrees:

```
>SXAGH021045<
```

Now a reporting event can be created using the *J* signal associated with the heading delta we just created:

```
>SED40NV0;J02+<
```

4.16 Driving Metrics (Acceleration, Max. Speed, etc)

Antares SBTM has the ability of calculating the vehicle's positive and negative accelerations based on GPS information. The instant acceleration value gives an idea of how a vehicle is being drove. Large positive values indicates excessive use of gas pedal and large negative values indicates hard-breaking

4.16. DRIVING METRICS (ACCELERATION, MAX. SPEED, ETC)

situations. A vehicle having large acceleration values indicates unsafe an expensive driving. On the contrary, low values indicate an smooth, efficient driving. The actual acceleration of the vehicle can be known with the **XAIT** command. It could also be programmed to be reported by using **QXAIT** on an event's user-action field or by using an extended-EV reporting messages which includes the acceleration value.

The **XAIT** command also gives information on the maximum positive acceleration, maximum negative acceleration and maximum speed since the last *Driving Metrics* reset. These values are stored along with the GPS information at the moment the maximum occurred (i.e. Date, time, position, speed, heading). A *Driving Metrics* reset is performed by the user with the same **XAIT** message. Notice that the unit could also be programmed to do this automatically by adding the reset command on an event's user-action field. For example, the *Driving Metrics* can be reset any time the vehicle's ignition is turned off and after sending the *Driving Metrics* of the last trip.

The acceleration is measured in Miles per Hour per Second. This is, how much the speed increases or decreases on a second basis. For example, if a vehicle goes from 0 to 32mph in 4 seconds, the resulting acceleration will be of $32/4 = +8$ Miles/(h*s). When a vehicle breaks, the resulting acceleration is measured in the same manner but it will be shown as negative. Typical 'good' values stay between -13 and +13, but what are 'good' and what are 'bad' values is something that has to be determined by each client in a comparative manner between good and bad drivers.

Driving Metrics information is consulted (and reset) via the **XAIT** command. See the TAIP chapter for detailed information. To query the actual driving metrics resume, send to the unit:

```
>QXAITR<
```

The unit responds something like:

```
>RXAITR+03+09-10,041;ID=EXAMPLE<
```

This particular reading indicates that the actual acceleration is +3Miles/(h*s) (or MPH*s), the maximum acceleration since the last *Driving Metrics* reset has been +9Miles/(h*s) and the maximum negative acceleration (maximum breaking) has been -10Miles/(h*s). The maximum registered speed is 41MPH.

Now, to know where the maximum breaking occurred, the GPS information for the maximum negative acceleration is consulted:

```
>QXAIT1<
```

The unit responds something like:

```
>RXAIT1-10EV001447147233+2578230-0802813901519512;ID=EXAMPLE<
```

Showing the max. negative acceleration of -10 and adding the underlined part which is an event 00 EV message with all of the GPS information at the moment the maximum breaking occurred.

To clear the maximum accelerations and maximum speed, the following

4.17. USING ACCELERATION SIGNALS

command is sent:

>SXAITC<

For detailed information consult the XAIT message in the TAIP chapter.

Acceleration limits can be set to trigger events whenever the acceleration exceeds given positive or negative thresholds. See the next section.

4.17 Using Acceleration *signals*

See the previous section for an introduction on how acceleration is calculated and used.

Positive and negative acceleration thresholds can be set to trigger an event and also to report a GPS Back Log to have a 1-second follow up of the vehicle, previous to a large acceleration condition. To do so, the Acceleration Signals (Nxx) are used. Up to 5 thresholds can be set with the XAGN message.

For example, to set a threshold on +13Miles/h*s and another threshold at -18Miles/h*s, send to the unit:

>SXAGN001+13<

>SXAGN011-18<

Now, to have the unit send event 05 whenever the positive threshold is exceeded (large gas pedal usage), and event 23 whenever the negative threshold is exceeded (large breaking condition or crash situation), use:

>SED05XV0;N00+<

>SED23XV0;N01+<

Warning:

Note that for positive acceleration thresholds, Acceleration Signals are TRUE when the vehicle's acceleration is larger than the threshold. For negative acceleration thresholds, Acceleration Signals are TRUE when the vehicle's acceleration is less than the threshold.

See the XAGN message for detailed information.

Now, to have the unit report a GPS back of the previous 60 seconds before a large breaking or crash condition, use the XAKL message in the event's user-action field:

>SED24XV0;N01+;ACT=QXAKL<

4.18. VOICE CALLS

This will make the unit send the last 60 GPS messages *Antares SBTM* has received from its GPS module.¹⁵ The last 15 seconds can be retrieved instead of 60, just change QXAKL to QXAKL0015. See the XAKL message and the *GPS Back Log* section for more information.

4.18 Voice calls

To make the *Antares SBTM* generate/terminate a voice call refer to the XAVC and XAVE messages.

To answer voice calls:

If no voice call is being held, the unit can answer incoming calls as long as the remote party number is registered on any Phone number-type *Destination* and the *Destination* is authorized to receive phone calls. If number identification service is not provided by the network operator, the authorization process can not take place and the incoming call is rejected. The unit rejects a call by hanging up immediately after answering.

Note:

Number-identification-service has to be enabled for the unit's line by the carrier in order for the unit to answer phone calls.

To determine the incoming voice call authorization on a given *Destination*, the C field on the Phone Number-type DP message is used. Refer to the XADP message for more information.

To monitor the state of a voice call use the voice signals:

- V00: Any voice call is taking place.
- V01: A voice call with DP 10 is taking place.
- V02: A voice call with DP 11 is taking place.
- V03: A voice call with DP 12 is taking place.
- V04: A voice call with DP 13 is taking place.
- V05: A voice call with DP 14 is taking place.

4.19 Battery monitoring

To monitor the internal back-up battery refer to the XABS message.

To generate events according to charge level changes use the battery signals (Bxx) and their definition with the XAGB message.

¹⁵ Depending on Hardware configuration this value does not lead necessarily to 60 seconds but to some value between 50 and 60 seconds, the GPS Back Log data must be analyzed to determinate this

4.20 Serial port devices

Depending on the type of data that it is going to be exchanged between the *Antares SBTM* and the external RS-232 device there are two possible configurations:

1. The external device talks TAIP so the main reason for connecting it with the unit is to receive GPS-like information and display it locally. An example is a laptop running an AVL application or a PDA running a GPS navigation software.
2. The external device has no idea of TAIP so it talks any byte-based protocol.

For the first option there are no special configurations on the *Antares SBTM*. It is only necessary to configure the *Event Machine* to send reports to the serial port so the attached device has GPS/events information to analyze.

For the second option, the *Antares SBTM* is used just as a communication media between the attached device and a *Destination*. When working in this mode the *Antares SBTM* can pass any byte-like message to a *Destination* and vice versa.

This requires the *Antares SBTM* working on MDT PAD mode. This is controlled by the MT messages. When working in this mode the TAIP console over the serial port is disabled. In order for the serial port to enable the TAIP console again, the user-defined escape character has to be entered. The default value for the escape character is (1B hex) or the 'ESC' key on a terminal.

Note:

If the serial port is not responding to TAIP commands it may be that the serial port is in MDT PAD mode. In this case the escape char has to be entered. The default escape char is **1B hex** or the 'ESC' key on a serial port terminal. Another option to disable the MDT mode is sending the string EXIT_COMMDATA as a single packet. This is also the only option to disable the Full Binary MDT mode through the serial port.

On MDT PAD mode *Antares SBTM* uses a PAD¹⁶ criteria to send the received message to a *Destination*.

The PAD criteria is controlled with the MT message and it is composed by:

- PAD character 1.
- PAD character 2.
- PAD Timeout.
- PAD size.

¹⁶ Packet Assembly Disassembly.

4.20. SERIAL PORT DEVICES

Whenever any of the PAD criterion is met, the accumulated message on the unit is transmitted.

The unit sends and receives the MDT messages using the TX TAIP message. Using this TAIP message any binary-like data can be transferred by means of escape characters. See the TX message on the TAIP reference for more information.

Example Having the unit to work on MDT PAD mode so that the serial port can exchange binary data with the IP-type *Destination* 04. The attached device uses the byte 03 to signal the end of a binary message.

For this example let's use the following PAD criteria:

- PAD character 1: 03 (not included on message)
- PAD character 2: FA
- PAD size: 50
- PAD timeout: 10 seconds.

Let us include the PAD characters on the message, and use as escape char the 1B hex value. Now we use the MT message to drive the unit to PAD mode passing the criteria we just created:

```
>SMTP010050T\03\FA\1B\FF;ROUTE=4<
```

After responding the command, the serial port is driven into PAD mode. For now on, binary data can be exchanged.

Let's see how data is transformed in order to be delivered to the remote destination and vice versa.

Supposing the external device sends the following data:

Note: Non-printable data is shown between [] symbols in hex value

```
[02]Test message...[03]
```

The *Antares SBTM* after receiving the 03 byte will send to *Destination* 04 the following TX message:

```
>RTX\02Test message...\03<
```

Now, if the device sends:

```
[02] [10] [1F] [AB] [AB] [3B] [12]ABCD [08] [09] [FC] [03]
```

The unit will send the TX message:

```
>RTX\02\10\1F\AB\AB\3B\12ABCD\b\t\FC\03<
```

Notice that for bytes 8 and 9 the special characters listed on the TX message were used.

The inverse process works in the same way. When the remote host sends to *Antares SBTM* a TX message it can contain any byte-like character by means of escape sequences.

4.21. ANALOG TO DIGITAL CONVERTER MONITORING

If the host would like to send the sequence:

[02] [07] [08] [FC] [AA]xyz[F1]

to the device attached on the serial port, then it has to send the following TX message to the *Antares SBTM* :

>STX\02\a\b\FC\AAxyz\F1<

This will make the *Antares SBTM* send the original sequence to the attached device.

There is also a Full Binary MDT mode, which allows to use the full range of binary characters. The Full Binary MDT mode does not have any user defined escape characters or any defined Packet delimiters. To enable the Full Binay MDT mode, use the H mode of the MT TAIP message, I.E: >SMTH< To exit from the Full Binary MDT mode send the TAIP message >SMTN< over the air, or the string EXIT.COMMDATA as a single packet over the serial port.

For more information refer to the TX and MT messages.

4.21 Analog to Digital Converter monitoring

The ADC can be monitored with the XAAC message.

To generate events according to ADC changes use the ADC signals (Dxx) and their definition with the XAGA message.

4.22 Using a TCP/UDP keep-alive

Refer to the XAKA message.

4.23 IMEI as ID

Antares SBTM can be programmed to use its International Mobile Equipment Identity (IMEI) as ID instead of a user-set value. This is useful for managing units without worrying of duplicate or changed IDs. It also eases the programming task as this number is already stored on every unit.

To instruct the unit to use its IMEI as ID instead of the ID set by the user with the ID command, use the XAID command in the following way:

>SX AID1<

This will make *Antares SBTM* use its IMEI as ID and it will ignore any value set with the ID message. To instruct the unit not to use its IMEI as ID, send to the unit:

>SX AID0<

Note that the unit uses the user-defined ID (ID command) by default. You can also consult the unit's IMEI with the XAIM command.

4.24 Cell ID reporting

Antares SBTM can add the Cellular Network Cell ID information on every reported message. This enables a Tracking System to locate the unit when GPS is not available. *Antares SBTM* will report the Cell ID, LAC, MCC, MNC and RSSI of the cell it is registered with. This information can be used by systems that know the location of Cells to approximate a location of an unit with no GPS.

There are two methods for making the unit report Cell ID information:

1. Use the extended EV tags ;CE or ;CF.
2. Use QXACE as an user-action on an event definition.

ext-EV method The first method is preferred as Cell ID information is attached to GPS information giving a clearer situation of the unit when the report generated.

For information on extended EV tags see the *Reports' messages* section on the *Operation* chapter, the *Scenarios and Examples* chapter and the XAEF and ED messages.

The extended-EV tag ;CF gives information about the actual Cell's:

- MCC: Mobile Country Code: 3 digits.
- MNC: Mobile Network Code: 3 digits
- LAC: Local Area Code: Cells are grouped by an operator in a Location Area group which is identified by this number between 0 and 65535. (Represented in HEX (0000-FFFF)).
- Cell ID: Cell Identity. Number between 0 and 65535. (Represented in HEX (0000-FFFF)).
- RSSI: Received signal strength indicator. Number between 0 and 63.¹⁷

The extended-EV tag ;CE gives information about the actual Cell's:

- Cell ID: Cell Identity. Number between 0 and 65535. (Represented in HEX (0000-FFFF)).
- RSSI: Received signal strength indicator. Number between 0 and 63.

To have the unit report CI and RSSI on every EV message, set the extended-EV format A to contain ;CE:

```
>SXAEEFA;CE<
```

Now every event which is defined with 'A' as message type will add the ;CE information:

```
>SED30N0;J00+<
```

¹⁷ See the EV message for detailed information

4.25. SLEEP MODE

Event 30 will be reported as:

```
>REV301447147509+2578250-0802813901519512;CE=233428;ID=EXAMPLE<
```

The reported message is explained in detail on the EV message description. See the TAIP messages' reference.

user-action method The second method for obtaining Cell ID reporting from the unit is to add a Cell ID query to an user-action field on an Event Definition.

The previous Event 30 could have been defined as:

```
>SED30NV0;J00+;ACT=QXACE1<
```

Thus generating a Cell ID query whenever J00 signal gets high (turn-by-turn report). After event 30 is reported, the query generates a report with the answer to the Cell ID query:

```
>RXACE1;208,00,1194,7ef1,27,739,9,,,2,,,0<
```

Also, a 6-neighbor query can be done:

```
>SED35NV0;U00+;ACT=QXACE2<
```

For detailed information see the XACE message.

Cell ID change reporting The Cell ID change signal F14 is included to generate a report whenever the unit moves from one Cell to another. This can be used by a Tracking System to create an Operator's Cell ID map by combining the Cell ID and GPS information reported by the unit whenever the Cell changes. Just create an event that depends on the Cell ID change signal and that uses the extended-EV format A defined on the previous example.

```
>SED00NA0;F14+<
```

4.25 Sleep mode

Refer to the XAPM message.

4.26 Restoring the unit

The drive the unit to its factory defaults use the >SRT;ALL< command. A system reset is automatically performed after receiving the command. You can not use this command over the air. You can also use the >SRT;CONFIG< to delete all parameters but a few essential communication ones. Refer to the RT message for more information.

4.27 Resetting the unit

Use the >SRT< command to perform a software reset on the unit.

4.28 Using Scripts

Scripts are text files with extension *.tmf* used to create/store/edit an *Antares SBTM* configuration.

These text files are passed to the unit via serial port¹⁸ with *TAIP DownloaderTM Tool*. See the *TAIP DownloaderTM Tool* section of the *Operation* chapter for information on using this software.

On the script file you put the same configuration messages that will be given manually to the unit in a new line each. The order of messages is not relevant as they are all going to be given to the unit at the moment of writing the script.

The file can contain user comments. These are user-defined texts that have no meaning for the unit. Comments are created to help reading or understanding the configuration messages. To create a comment start a new line with the hash symbol (#) and write ANY TEXT THAT DOES NOT CONTAIN THE > OR < CHARACTERS. For example:

```
#This is a valid comment
```

```
#This <is not> a valid comment!
```

If one line is not enough for the comment a new comment line has to be created, for example:

```
#This is a long comment that  
#spans to more than one line of text.
```

Comments are not passed to the unit, they are only saved on the *.tmf* file and can not be retrieved when reading an script from an unit.

Here it is an example of an script file:

```
#Antares SB script  
#Getting Started example  
  
#Unit's ID  
>SIDEEXAMPLE<  
  
#configuring the SIM's PIN  
>SRFI1234<  
  
#configuring the APN  
>SRFAinternet.carrier-name.com<  
  
#The remote AVL server address and port  
>SXADP0000avl.server.com;2145<  
  
#A Destination Address holding the
```

¹⁸ By using third-party softwares, *TAIP DownloaderTM* can pass/read an entire script over the air.

4.28. USING SCRIPTS

```
# server destination
>SDA4;P00<

#Time-only Time And Distance
#signal definition
>STD80300<

#Event triggered by T&D signal
>SED37NV4;TD8+<

#Input report event
>SED05NV4;IP3+<

#end
```

4.28.1 Creating an script from scratch

Creating an script is not different than giving the configuration commands manually to the unit.

Create a new empty file on any low-level¹⁹ text editor like WindowsTM, notepad. Start adding a new line for each configuration message, when finish, save the file as a `tmf` file.

Note for notepad users: Notepad won't let you save the file with an extension different to `txt`, to be able to save it with the `tmf` extension make sure that on the **Type** box of the **Save as** dialog, **All files** is selected, then enter a name followed by `.tmf`.

Now the `tmf` file you just created can be passed to an unit using the *TAIP Downloader*TM software, see the *Writing Scripts* section for this.

If you prefer, you can edit an existing script instead of creating a new one from scratch.

Another option for not having to create an script from scratch is making *TAIP Downloader*TM generate an script file by reading an *Antares SB*TM. See the next section.

4.28.2 Reading Scripts

You can read an *Antares SB*TM's configuration and save it on a `tmf` file with the *TAIP Downloader*TM software. This is useful to read, edit or duplicate (i.e. copy the configuration to other units) a unit's configuration.

Follow the instructions on the *TAIP Downloader*TM *Tool* section of the *Operation* chapter and have *TAIP Downloader*TM already connected and communicating with *Antares SB*TM before proceeding.

¹⁹ Do not use high-level editors like word processors as they will add confusing no-plain-text characters to the script.

Warning:

Make sure of using *TAIP DownloaderTM* version 1.0.2 or superior. Contact *Digital Communications TechnologiesTM* for information on how to upgrade.

Using *TAIP DownloaderTM* go to the Device menu and select the “Read configuration” option. A file name and location will be asked. After clicking the “Save” button the reading process starts. This process takes approximately 1 minute and depends on the unit’s configuration. While the reading process is taking place you will see the dialog: “Sending Message” with a cancel option. If you get an error when reading see the TAIP error list on the TAIP specification chapter.

Now you can view/edit the generated tmf file and check the unit’s configuration.

4.28.3 Writing Scripts

Make sure of following the instructions on the *TAIP DownloaderTM Tool* section of the *Operation* chapter and have *TAIP DownloaderTM* already connected and communicating with *Antares SBTM* before proceeding.

Warning:

Make sure of using version 1.0.2 or superior of *TAIP DownloaderTM*. Contact *Digital Communications TechnologiesTM* for information on how to upgrade.

On *TAIP DownloaderTM* go to the Device menu and select the Write configuration option. The software will ask for a tmf file to use, select the tmf file that holds the configuration script, after clicking Open, the writing process starts.

Refer to the TAIP specification error list for any error on the process.

4.28.4 Scripts Over The Air

TAIP DownloaderTM Tool uses the PC’s serial ports to communicate with a unit. By means of third party softwares a virtual serial port can be created, so that the virtual communication is done via TCP or UDP. This enables the reading and writing process of scripts to be done remotely.

One of such tools that creates a virtual serial port connected through a TCP connection is HW Virtual Serial Port which can be found at:

<http://www.HW-group.com>

Remember that the *Antares SBTM* works as a TCP client so HW Virtual Serial Port has to be configured as server. To do so make sure to select the option HW VSP works as the TP Server only box on the software.

4.28. USING SCRIPTS

For more information on these software tools contact *Digital Communications TechnologiesTM*.

5 Scenarios and examples

For this chapter It is assumed that the reader already has access to the unit's TAIP console through the unit's serial port using a terminal software like Windows[™], Hyperterminal[™]. Refer to the *Operation* chapter for more information.

5.1 Getting Started

This example is a step by step instructional on getting started with the unit's configuration. At the end the reader should have the unit reporting an input activation and a periodic message to an IP host.

5.1.1 Setting the unit's ID

In order for the unit to be identified on an AVL server an ID has to be set. The default value for the unit's ID is 0000. This field may be any string of 10 characters maximum.

For this example let's call our unit **EXAMPLE**:

```
>SIDEEXAMPLE<
```

5.1.2 Setting the APN and PIN

After configuring an ID, the first step is enabling the unit on the GSM/GPRS network. For this an APN provided by the cellular carrier is required and depending on the SIM card configuration a PIN value will be required too. For this example we will use APN **internet.carrier-name.com** and PIN 1234.

Setting the SIM card's PIN:

```
>SRFI1234<
```

If your SIM card does not require a PIN you can omit this step.

Setting the APN:

```
>SRFAinternet.carrier-name.com<
```

At this point the unit will try to register on GSM and on GPRS.

Note:

Although PIN and APN parameters take effect immediately the unit may take up some time on registering to the network if a previous erroneous PIN was used. You can wait for the unit to register or you can speed up the process by resetting it with the **>SRT<** command.

The GSM registration status is indicated by the *status* LED. This LED is solid whenever the unit is not registered on GSM. Although the unit is not registered, it could be searching for the operator which is a normal condition. Other un-registered conditions may be detected by means of the RP message. When the unit is registered on GSM the *status* LED should be either blinking or completely OFF. In this state the LED shows the received signal strength. Refer to the *Operation* chapter for more information on LEDs and to the TAIP reference for the RP message.

The GPRS session status is shown by the *On line* LED. The unit may only be registered on GPRS whenever it is GSM-registered. When the GPRS session is up and ready the *On line* LED is either blinking or solid. Although this LED being OFF indicates that the GPRS session is not ready it could mean that the unit is trying to establish the session and it could already be GPRS-Attached¹. A blinking *On line* LED indicates that GPRS registration is OK, but shows that there are problems communicating with a remote IP server which is a non-GPRS related problem.

5.1.3 Creating a Destination Point (DP)

Now that we have the unit working on the GSM/GPRS network the unit is ready to send and receive communications from IP servers and phone numbers. For our example we need to create a *Destination Point* (DP) which holds our remote server IP number or address and the serving port that it is using for listening to TCP connections or UDP datagrams. As discussed on the *Operation* a *Destination Point* can work either with TCP or UDP protocols, you can change this with the 'C' flag of the *Destination Point* configuration, see the XADP message. For this example we will work with a remote server that listens for TCP connections.

If you do not already have an AVL server listening for TCP connections see the *Operation* chapter before continuing.

For this example we will use a server located on the address `av1.server.com` which listens for TCP connections on the port 2145. If we want to use name addresses instead of IP numbers we have to use the XADP message instead of the deprecated DP message. With the XADP message we can also specify an IP number so the use of DP is obsolete and is maintained only for backward compatibility.

The *Destination Point* index we chose is only restricted in our example by the DPs destined to be IP hosts (00 to 09). We chose then DP 00 for no other special reason:

```
>SXADP0000av1.server.com;2145<
```

An IP address could also be specified. Supposing we wanted to use the IP number 192.168.0.1 we would have to send:

¹ Being attached to the GPRS network is not the same as having completed the GPRS session start-up.

5.1. GETTING STARTED

>SXADP0000192.168.0.1;2145<

This last option with the deprecated message would have been:

>SDP000019216800000102145<

Having set the DP the unit will automatically start opening a TCP connection with the server (as long as GPRS is ready) *even if it has no messages to send to it*. This is a programmed feature of the unit that makes it (re)open the TCP connection whenever the network is available after being down or whenever the connection gets closed. At this point is possible for the *On line* led to start blinking, meaning that for whatever reason the TCP connection with any of the configured IP-type DP is not open.

If after configuring the DP the *On line* LED remains solid, the unit has established the TCP connection with the server and it is ready to send or receive messages to/from it. If on the contrary the LED starts blinking some considerations have to be taken:

1. The AVL software server is not running or it is running but it is not listening for TCP connections.
2. The listening port and/or address is wrong.
3. The server is behind a firewall/router/NAT that prevents the incoming connection from passing to the TCP listener.
4. The server is accepting the connection but it is immediately (or a few seconds later) closing it.
5. The *Antares SBTM* is behind a cellular carrier's NAT which has the selected port blocked.
6. The selected APN has no Internet access. Or in case of a private network, the APN has no access to the network where the AVL server is running.
7. There are network related problems that prevent the unit from communicating even with GPRS up.

5.1.4 Creating a Destination Address (DA)

As mentioned on the *Destinations* section a DA has to be created so that an event's routing option can be completed. In our example we only have to create a DA with a single *Destination Point* which is the one we just created. We have no restrictions for the DA range (0-9) so we chose DA 4 for no special reason:

>SDA4;P00<

Indicating that *Destination Address* 4 is the grouping of the single *Destination Point* 00.

5.1.5 Creating a time-period criterion

For this example we want the unit to send a report based on a time-only criterion which will make the unit send a reporting message every x elapsed

5.1. GETTING STARTED

minutes. There are several ways of doing this but one of the most common is to configure a *Time And Distance* signal with no *Maximum Time Between Reports* and no *Distance Threshold* parameters so it triggers a TD signal on a time-only basis set by the *Minimum Time Between Reports* parameter. Refer to the TD message for more information.

Let's use a reporting period of 5 minutes (300 seconds). For no special reason let's choose TD signal 8 to do the job:

```
>STD80300<
```

This will make the unit activate signal TD8 every 5 minutes so we can create an event triggered by this signal which is going to generate the periodic report.

Note that in order to keep this example simple, we are using a basic time-only report, but this approach is not advised on a real world scenario where a vehicle remains at rest most of the time and where having a time-only criterion will generate a bunch of unnecessary messages. It is recommended to use the three parameters of the *Time And Distance* definition to achieve a more intelligent report.

5.1.6 Tiding a signal to an event

With the signal TD8 generating a pulse every 5 minutes the only thing left to do is defining an event that triggers with this condition. At this point we need to ask ourselves what event code to chose and what kind of message send to the AVL server. The answer lies on the AVL server configuration: The event code has to have any meaning for the AVL software and the type of message depends on the kind of information we will like to get from the unit's report. There are two kind of messages, the EV gives more information than the ET which only gives the time and date when the event occurred. In order to receive full GPS information we need to use the EV report.

As event code (00-49) we chose for no special reason code 37:

```
>SED37NV4;TD8+<
```

Notice we are using DA 4. This will make the report generated by event 37 to be sent to the single DP 00 which is our AVL server.

For more information consult the ED message on the TAIP reference.

5.1.7 Checking the host software/server

A this point and as long as the *On line* LED remains solid, an EV message should be arriving to the AVL server every 5 minutes. The software has to be able to interpret TAIP EV and/or ET messages. If you run into troubles checking your AVL application you can always shut it down an use some popular free TCP listener applications that will show you the RAW data where you can check the TAIP messages generated by the *Antares SBTM*. Other useful tool is a network sniffer that allows you to analyze the traffic on a given TCP/UDP connection while you run the AVL server.

5.1. GETTING STARTED

Some examples of such tools can be found at:

- TCP client and listener: **TCP Test Tool**[™] from www.simplecomtools.com
- UDP client and listener: **UDP Test Tool**[™] from www.simplecomtools.com
- Network sniffer: **Wireshark**[™] from <http://www.wireshark.org/>

5.1.8 Adding an Input report

Now we will like to create an Input report to the AVL server having the event code 05, whenever the Input 3 goes high. This is a simple event that depends on a single signal transition, signal IP3 which is the same as signal G03²:

```
>SED05NV4;IP3+<
```

Now our unit is generating a 5 minutes periodic report and also a special report whenever the Input 3 goes high³.

5.1.9 Script

This example can be resumed with the following script:

```
#Antares SB script
#Getting Started example

#Delete any previous configuration
>SRT;CONFIG<
>SXADP**U<

#Unit's ID
>SIDEEXAMPLE<

#configuring the SIM's PIN
>SRFI1234<

#configuring the APN
>SRFAinternet.carrier-name.com<

#The remote AVL server address and port
>SXADP0000avl.server.com;2145<

#A Destination Address holding the
```

² G signals are maintained for backward compatibility. The use of IP and XP signals is preferred.

³ An input high is an input at GND or 0 volts.

5.2. ADDING SMS REPORTING

```
# server destination
>SDA4;P00<

#Time-only Time And Distance
#signal definition
>STD80300<

#Event triggered by T&D signal
>SED37NV4;TD8+<

#Input report event
>SED05NV4;IP3+<

#end
```

You can copy and paste this script to a new empty text file and have it save with a .tmf extension so *TAIP Downloader*TM software can pass it to a unit. See the *Using Scripts* section of the *Operation* chapter for information on creating, editing and downloading scripts.

5.2 Adding SMS reporting

Based on the *Getting Started* example, SMS reporting is added.

On this example we are going to make the unit report the previously defined Input report (IP3) to a Phone Number as well as to the remote AVL server with a custom text message.

5.2.1 Create the SMS Destination Point

The first step is configuring the destination Phone Number on one of the unit's Phone-Number-Type *Destination Point* (DPs 10 to 14). For no particular reason we choose DP 10. The number we are going to configure on DP 10 is 912345678. See an special note about some Phone Numbers' variations on the *Over The Air* section of the *Operation* chapter.

```
>SXADP1010912345678<
```

Refer to the **XADP** message for more information on the command.

Notice the underlined "1": With this, we are indicating that events' messages sent to DP 10 are always going to be user-defined texts (not TAIP EV messages). The "0" next to the underlined "1" is not relevant for this example but it is going to be important for the next one as we would like the Phone Number to have TAIP-console and voice access.

5.2.2 Create a new *Destination Address*

For now we only have DA 4 that associates only DP 00. We need to create another DA that includes our AVL server (DP 00) and the recently created Phone Number (DP 10). For no particular reason we choose DA 5 to do the job:

```
>SDA5;P00,P10<
```

5.2. ADDING SMS REPORTING

Now *Destination Address* 5 holds *Destination Points* 00 and 10. We can point any event to DA 5 making any report generated by the event to be routed to our AVL server and to the Telephone Number 912345678 at the same time.

5.2.3 Change the Input report event definition

Our previously input report event was defined on event 05. We are going to use the same definition but as you will notice we are going to change on the event's routing options the selected DA from 4 to 5:

```
>SED05NV5;IP3+<
```

Now any time input 3 goes high (IP3+) an event report is going to be generated to both the AVL server and the Phone Number 912345678.

5.2.4 Create a SMS custom message

The only thing left to do is create a user-defined text to be sent to the Phone-Number-type destination whenever event 03 occurs. For this we use the XATM message which allows to create up to 50 user-messages, one for each event. Refer to the XATM message on the TAIP reference for more information. Let's use the text **Input 3 has been activated!**:

```
>SXATM05Input 3 has been activated!<
```

The underlined 05 indicates that the message we are defining is for event 05.

5.2.5 Check the reported message

The following text message should be arriving to the Phone Number 912345678 any time input 3 goes high:

```
Antares:EXAMPLE
Event:05
Input 3 has been activated!
```

If no user-message is defined with the XATM message the following message is sent:

```
Antares:EXAMPLE
Event:05
-----
```

If on the *Destination Point* we change the underlined "1" for a "0" we receive an EV message on the Phone Number instead of the user-defined text, something like:

```
>REV050000001234+0000000+00000000000000001;ID=EXAMPLE<
```

At the same time input 3 goes high an EV message is also sent to the remote AVL server. Notice that we did not change the definition of the periodic report event (event 37), so the periodic report is still being sent to the AVL server only.

5.2. ADDING SMS REPORTING

If no SMS is received on the Phone Number-type destination, consider the following:

- The unit can only send SMS messages when registered on GSM. The *Signal* LED does not have to be solid ON.
- The cellular carrier has to provide SMS sending capabilities to the cellular line being used by the *Antares SBTM*.
- The phone number 912345678 may not be exactly the phone number that should be used for sending SMS messages. A plus sign with country/area code or similar may be required. See the note on the *Over The Air* section of the *Operation* chapter.

5.2.6 Script

This example which is based on the Getting Started example is resumed on the following script:

```
#Antares SB script
#Getting Started example
# + SMS reporting

#Delete any previous configuration
>SRT;CONFIG<
>SXADP**U<

#Unit's ID
>SIDEEXAMPLE<

#configuring the SIM's PIN
>SRFI1234<

#configuring the APN
>SRFAinternet.carrier-name.com<

#The remote AVL server address and port
>SXADP0000avl.server.com;2145<

#The Phone-Number-type destination
#with support for user-messages ON
>SXADP1010912345678<

#A Destination Address holding the
# server destination
>SDA4;P00<
#A Destination Address holding the
# server destination and phone number
>SDA5;P00,P10<

#Time-only Time And Distance
```

5.3. ADDING SMS INTERACTION

```
#signal definition
>STD80300<

#Event triggered by T&D signal
>SED37NV4;TD8+<

#Input report event
>SED05NV5;IP3+<

#User text message for event 05
>SXATM05Input 3 has been activated!<

#end
```

5.3 Adding SMS interaction

Once a Phone-Number-type destination has been created it is possible to interact with the unit via SMS messages generated from that phone number, as long as the *Destination* configuration allows TAIP console access to the configured DP.

This means taking care of the *Access* parameter on the *Destination Point* definition. Continuing with our previous example. We defined the Phone-Number-type DP 10 as:

```
>SXADP1010912345678<
```

Refer to the **XADP** message for more information. The underlined value indicates the access parameter for this DP, indicating that the Phone Number 912345678 has TAIP console access via SMS messages.

Now that we are clear on the access level of the Phone-Number-type DP, interacting with the unit is a matter of sending TAIP commands through SMS messages and waiting for response as incoming SMS texts.

5.3.1 Query the unit with a SMS

To ask for the current position-velocity solution of the unit, sent a from the Phone Number 912345678 an SMS with the query:

```
>QPV<
```

The unit will respond with an SMS containing the text:

```
>RPV00123+0000000+0000000000112012<
```

5.3.2 Set an output with a SMS

Driving outputs is achieved by using the *Set* form of the *SS* message. To set output 2 high via a SMS message, send to the unit from the authorized

5.4. ADDING VOICE INTERACTION

phone number 912345678 the text message:
>SSSXP21<

The unit responds the following via SMS:
>RSSXP21<

Confirming that it accepted the command and the output has been set high.

5.4 Adding voice interaction

The unit may receive or make calls from/to any of the Phone Number-type defined *Destinations*. To initiate a voice call to a DP the **XAVC** message is used. To receive voice calls the remote Phone Number has to be defined on a Phone-Number-type destination and the *Access* parameter on the DP definition has to indicate that an incoming call from that DP is to be answered.

5.4.1 Make the unit accept a phone call

To do so the *Access* parameter on the corresponding DP has to indicated voice call access. In our previous example the 912345678 Phone Number was defined with the **XADP** messages as follows:
>SXADP1010912345678<

The *Access* is such that voice calls from that number are to be answered.

Any incoming call from the 912345678 is now going to be answered. The voice call is maintained until the 912345678 decides to hung-up.

For more information on the *Access* parameter see the **XADP** message.

5.4.2 Have the unit initiate a voice call

This can be done manually by using the **XAVC** message.

To initiate a voice call to the Phone Number defined on the *Adding SMS reporting* example send to the unit:
>SXAVC10<

The unit returns immediately:
>RXAVC10<

Indicating it has accepted the command and that the voice call dial up has initiated. No failure/success indication is thrown at a later time. To monitor the voice call the fixed voice signals have to be used.

To end the voice call, send:
>SXAVE<

5.5. IGNITION DETECTION

The unit can also be programmed to start the voice call automatically by using an event's user-action field. Continuing with our previous example, the event 05 definition can be altered to include the voice call to DP 10:
>SED05NV5;IP3+;ACT=SXAVC10<

This will make the unit start a voice call to Phone Number 912345678 any time the input 3 goes high.

5.5 Ignition detection

Generating an Ignition ON and/or and Ignition OFF event is not different than working with any other input, you just have to create event's related to the ignition sense signal, the F00.

Continuing the *Adding SMS reporting* example, to generate an event code 19 for an Ignition ON send to the unit:
>SED19NV4;F00+<

And event code 20 for an Ignition OFF situation:
>SED20NV4;F00-<

This last one could have been defined as:
>SED20NV4;F00!+<

Resulting in the same performance.

Both events' DA is DA 4, the same used by the periodic report event, this means that the Ignition ON and OFF events are only going to be sent to the AVL server.

5.5.1 Script

The resulting script:

```
#Antares SB script
#Getting Started example
# + SMS reporting (+SMS interaction + Voice interaction)
# + Ignition sense

#Delete any previous configuration
>SRT;CONFIG<
>SXADP**U<

#Unit's ID
>SIDEEXAMPLE<

#configuring the SIM's PIN
>SRFI1234<

#configuring the APN
>SRFAinternet.carrier-name.com<
```

5.6. SPEED VIOLATION (WITH WARNING) REPORT

```
#The remote AVL server address and port
>SXADP0000avl.server.com;2145<

#The Phone-Number-type destination
#with support for user-messages ON
>SXADP1010912345678<

#A Destination Address holding the
# server destination
>SDA4;P00<
#A Destination Address holding the
# server destination and phone number
>SDA5;P00,P10<

#Time-only Time And Distance
#signal definition
>STD80300<

#Event triggered by T&D signal
>SED37NV4;TD8+<

#Input report event
>SED05NV5;IP3+<
#User text message for event 05
>SXATM05Input 3 has been activated!<

#Ignition ON event
>SED19NV4;F00+<
#Ignition OFF event
>SED20NV4;F00-<

#end
```

5.6 Speed violation (with warning) report

An speed limit violation event that is generated whenever the vehicle exceeds a determinated value can be configured as it was done on the *Setting Speed Limits* of the *Configuration* chapter. Here we are going to add a continuous-time condition with a visual warning (with a LED) so the driver has a chance to slow down before actually reporting the violation.

For this example let's set an speed limit of 65mph. Let's give 15 seconds to the driver for slowing down. Let's assume the indicator LED for the speed excess warning is wired to output 1 (XP1). The reporting event code for the violation, for no particular reason is 10.

5.6. SPEED VIOLATION (WITH WARNING) REPORT

5.6.1 Setting the speed limit

Up to 10 speed limits can be defined on the unit, for no particular reason we choose the limit 07. Using the **GS** message we set the limit:

```
>SGS0710650<
```

Now any time the vehicle speed is above 65mph the **S07** signal is going to be true.

5.6.2 Start a counter

In order for the violation to be reported only after a continuous-15-seconds condition, a timer has to be initiated whenever the speed goes above 65mph. To do so we create a silent event, that is it an event which only function is to start a timer but not to send any report:

```
>SED40SV0;S07+;ACT=SGC00TC00015<
```

We used event 40 for no particular reason other than it is free to be used.

The event triggers whenever **S07** signal changes to **true**, that is it, whenever the vehicle's speed goes beyond 65mph. The event includes an user-action: Starting timer 00 with a threshold value of 15 seconds. This will make signal **C00** go true 15 seconds after the counter definition, that is it, 15 seconds after the speed violation is detected.

5.6.3 Creating the violation report

Having **C00** signal become true 15 seconds after the speed violation the only thing left to do is create the violation event, the event 10:

```
>SED10NV4;C00+<
```

This one as you can see is not a silent event: It sends a report to **DA 4** which according to the example we have been working points to the remote AVL server.

5.6.4 Something is missing...

So far so good, it does what it is asked. But we are missing a silent event which turns off the counter whenever the driver slows down before the 15 seconds, or else the violation is going to be reported even though the driver did the right thing. To do so we create another silent event which undefines the **C00** counter whenever the speed falls below the limit:

```
>SED41SV0;S07-;ACT=SGC00U<
```

This will guarantee that the counter won't activate signal **C00** if the speed falls below the limit before 15 seconds.

5.6.5 Driving the LED

In order to lit LED connected to output 1 when the actual speed is greater than the speed limit, we create a silent event that drives output 1 high when the speed goes beyond the limit:

5.6. SPEED VIOLATION (WITH WARNING) REPORT

```
>SED42SV0;S07+;ACT=SSSXP11<
```

In a similar way as it was done with the 15 seconds counter, we have to create another silent event that turns the LED off when the speed limits falls below:

```
>SED43SV0;S07-;ACT=SSSXP10<
```

5.6.6 Script

The resulting script:

```
#Antares SB script
#Getting Started example
# + SMS reporting (+SMS interaction + Voice interaction)
# + Ignition sense
# + Speed viol. and warning
```

```
#Delete any previous configuration
>SRT;CONFIG<
>SXADP**U<
```

```
#Units ID
>SIDEEXAMPLE<
```

```
#configuring the SIMs PIN
>SRFI1234<
```

```
#configuring the APN
>SRFAinternet.carrier-name.com<
```

```
#The remote AVL server address and port
>SXADP0000avl.server.com;2145<
```

```
#The Phone-Number-type destination
#with support for user-messages ON
>SXADP1010912345678<
```

```
#A Destination Address holding the
# server destination
>SDA4;P00<
```

```
#A Destination Address holding the
# server destination and phone number
>SDA5;P00,P10<
```

```
#Time-only Time And Distance
#signal definition
>STD80300<
```

```
#Event triggered by T&D signal
```

5.7. START/STOP MONITORING

```
>SED37NV4;TD8+<

#Input report event
>SED05NV5;IP3+<

#User text message for event 05
>SXATM05Input 3 has been activated!<

#Ignition ON event
>SED19NV4;F00+<

#Ignition OFF event
>SED20NV4;F00-<

#Set an speed limit of 65mph
>SGS0710650<

#Start a 15 secs. timer when
#speed limit is violated
#(silent event)
>SED40SV0;S07+;ACT=SGC00TC00015<

#Stop the timer when the speed falls
#below the limit
#(silent event)
>SED41SV0;S07-;ACT=SGC00U<

#Create the violation report
>SED10NV4;C00+<

#Drive a LED on output 1 ON
#when the speed goes beyond the limit
#(silent event)
>SED42SV0;S07+;ACT=SSSXP11<

#Drive a LED on output 1 OFF
#when the speed falls off the limit
#(silent event)
>SED43SV0;S07-;ACT=SSSXP10<

#end
```

5.7 START/STOP monitoring

In this example we are going to create two reporting events that reflect two possible situations for a vehicle: a vehicle STOP and a vehicle re-start or first start (simply START). *Note: This example continues working on the example that has been worked so far, the Getting Started + SMS reporting*

5.7. START/STOP MONITORING

+ Ignition sense + Speed violation and warning.

Now we have to tell the unit what is a STOP and what is a START. There are several ways of doing this, for this example we are going to define a STOP situation to be when the vehicle's speed remains below a small speed limit for a given period of time. Having this, we define the START situation as happening when the vehicle speed's goes beyond the same limit and having the unit on a previous STOP condition.

In this way we can assure that a STOP is not going to be reported any time the vehicle's speed falls below a low value but that a time-condition is also required. We require the following two parameters:

1. An speed limit for determinating an STOP/START. This has to be a low value but not so low that the vehicle has the chance of moving extremely low and not having the reports.
2. A time condition. This value depends on what is an STOP situation for us. A vehicle may be at rest waiting for a traffic light to change and we may not want that situation to be reported as an STOP.

For our example let's use some typical values that work on the majority of configurations. For speed limit let's use 8mph and as time condition 2 minutes. For our example let's use for no particular reason event 30 for an STOP situation and event 31 for the START.

5.7.1 Setting a low speed limit

Using the GS signal we configure the 8mph limit:
>SGS0310080<

We chose speed limit 03 for no particular reason other than being free to be used.

5.7.2 Start a counter

In a similar way as it was done on the *Speed Violation* example we now create a silent event that starts a 2 minutes timer counter whenever the vehicle's speed falls off (opposite to the speed violation exp) 8mph:
>SED38SV0;S03-;ACT=SGC01TC00120<

Note that we use counter 01 which is not being used by any other event and event 38 that it is also free. Now any time the speed signal S03 gets false timer-counter 01 will start.

5.7.3 Create the STOP report

This event should trigger when the 2 minutes timer reaches its threshold value. That is it, when C01 becomes true, so we define the STOP event as:
>SED30NV4;C01+<

Notice that the event uses DA 4 and it is not silent, so the event is reported to the AVL server.

5.7.4 Create the START report

We define this event as occurring any time the vehicle's speed goes beyond the set limit of 8mph and a previous STOP condition existed. To do so we create the START event as depending on the S03 signal AND on the counter 01 signal C01. This works because once the C01 counter reaches the 2 minute threshold the C01 signal stays high "forever" (until the counter gets undefined by something else).

```
>SED31NV4;S03C01&+<
```

5.7.5 Something's missing...

As with the *Speed Violation* example we also need a silent event that undefines the counter in case the vehicle resumes its march before the 2 minutes condition, to do so we use silent event 39:

```
>SED39SV0;S03+ACT=SGC01U<
```

This assures that a vehicle's stop that lasts less than 2 minutes is not going to be reported.

5.7.6 Script

The corresponding script of this configuration plus all the examples worked so far is:

```
#Antares SB script
#Getting Started example
# + SMS reporting (+SMS interaction + Voice interaction)
# + Ignition sense
# + Speed viol. and warning
# + START/STOP monitoring

#--Delete any previous configuration-----
>SRT;CONFIG<
>SXADP**U<
#-----

#--Essential configurations-----
#Units ID
>SIDEEXAMPLE<

#configuring the SIMs PIN
>SRFI1234<

#configuring the APN
>SRFAinternet.carrier-name.com<

#The remote AVL server address and port
>SXADP0000avl.server.com;2145<
```

5.7. START/STOP MONITORING

```
#The Phone-Number-type destination
#with support for user-messages ON
>SXADP1010912345678<

#A Destination Address holding the
# server destination
>SDA4;P00<

#A Destination Address holding the
# server destination and phone number
>SDA5;P00,P10<

#Time-only Time And Distance
#signal definition
>STD80300<

#Event triggered by T&D signal
>SED37NV4;TD8+<
#-----

#--Input sense-----
#Input report event
>SED05NV5;IP3+<
#User text message for event 05
>SXATM05Input 3 has been activated!<
#-----

#--Ignition sense-----
#Ignition ON event
>SED19NV4;F00+<

#Ignition OFF event
>SED20NV4;F00-<
#-----

#--Speed viol. + warning-----
#Set an speed limit of 65mph
>SGS0710650<

#Start a 15 secs. timer when
#speed limit is violated
#(silent event)
>SED40SV0;S07+;ACT=SGC00TC00015<

#Stop the timer when the speed falls
#below the limit
#(silent event)
>SED41SV0;S07-;ACT=SGC00U<
```

5.8. SAFE ENGINE TURN OFF

```
#Create the violation report
>SED10NV4;C00+<

#Drive a LED on output 1 ON
#when the speed goes beyond the limit
#(silent event)
>SED42SV0;S07+;ACT=SSSXP11<

#Drive a LED on output 1 OFF
#when the speed falls off the limit
#(silent event)
>SED43SV0;S07-;ACT=SSSXP10<
#-----

#--START/STOP monitoring-----
#Speed limit for START/STOP detection
>SGS0310080<

#Start a 2 minutes timer
>SED38SV0;S03-;ACT=SGC01TC00120<

#Stop the timer
>SED39SV0;S03+ACT=SGC01U<

#Create the STOP report
>SED30NV4;C01+<

#Create the START report
>SED31NV4;S03C01&+<
#-----

#end
```

5.8 Safe engine turn off

This configuration is used to drive an output high after a user command⁴ only when the following condition is met:

The vehicle stays under a given low speed limit for a given period of time.

This is suitable for connecting an ignition cutter to the output so a vehicle's engine is only turned off when a safe speed condition exists.

For this example let's set an under-speed condition of 15mph with a time condition of 16 seconds. Let's assume the ignition cutter is wired to output

⁴ A command received locally or over the air.

5.8. SAFE ENGINE TURN OFF

4. We can also add an event to report when the output is actually set (i.e. the ignition is cut), let's use event code 13 for this. *We continue working on the same example worked so far.*

5.8.1 Create the speed limit

We set 15mph on the S01 limit:
>SGS0110150<

5.8.2 Creating a timer

We only want to cut the engine after the unit is instructed to do so. To do this we use a *User Signal*. This type of signal gets high or low whenever we tell it to. For this example let us use user signal 00 (U00). Now we create a silent event that starts a 16 seconds timer whenever the vehicle's speed falls off 15mph AND the user signal is set, that it is, when someone activates the user signal to indicate that the safe engine turn off process is to start:

>SED14SV0;S01!U00&+;ACT=SGC02TC00016<

We use event 14 and counter 02 for no particular reason.

5.8.3 Cutting the ignition

Now we create a non-silent event that sets the ignition OFF when the previous set counter reaches its threshold value of 16 seconds:

>SED13NV5;C02U00&+;ACT=SSSXP41<

The DA of this event is 5 meaning that the report of ignition cut-off will be sent to the AVL sever and the Phone Number 912345678.

Now any time someone activates the user signal 00 and after a low-speed + time condition, event 13 will set output 4 ON cutting the ignition of the vehicle. But as with the previous examples something is missing.

5.8.4 Stopping the counter

As it was done on the other examples, we need to stop the counter whenever the speeds goes beyond the limit, this to assure that if the time/speed condition is violated the engine won't be turned off:

>SED15SV0;S01+;ACT=SGC02U<

5.8.5 Restore the user signal

The only thing left to do is to restore the user signal so any time later when the ignition is restored the safe engine turn off mechanism is not triggered again without any user command.

>SED16SV0;E13+;ACT=SSSU000<

5.8.6 Script

Adding a safe-engine-turn-off option leads to the following script:

```
#Antares SB script
#Getting Started example
# + SMS reporting (+SMS interaction + Voice interaction)
# + Ignition sense
# + Speed viol. and warning
# + START/STOP monitoring
# + Safe engine turn off

#--Delete any previous configuration-----
>SRT;CONFIG<
>SXADP**U<
#-----

#--Essential configurations-----

#Units ID
>SIDEEXAMPLE<

#configuring the SIMs PIN
>SRFI1234<

#configuring the APN
>SRFAinternet.carrier-name.com<

#The remote AVL server address and port
>SXADP0000avl.server.com;2145<

#The Phone-Number-type destination
#with support for user-messages ON
>SXADP1010912345678<

#A Destination Address holding the
# server destination
>SDA4;P00<

#A Destination Address holding the
# server destination and phone number
>SDA5;P00,P10<

#Time-only Time And Distance
#signal definition
>STD80300<
```


5.8. SAFE ENGINE TURN OFF

```
#Event triggered by T&D signal
>SED37NV4;TD8+<
#-----

#--Input sense-----
#Input report event
>SED05NV5;IP3+<
#User text message for event 05
>SXATM05Input 3 has been activated!<
#-----

#--Ignition sense-----
#Ignition ON event
>SED19NV4;F00+<

#Ignition OFF event
>SED20NV4;F00-<
#-----

#--Speed viol. + warning-----
#Set an speed limit of 65mph
>SGS0710650<

#Start a 15 secs. timer when
#speed limit is violated
#(silent event)
>SED40SV0;S07+;ACT=SGC00TC00015<

#Stop the timer when the speed falls
#below the limit
#(silent event)
>SED41SV0;S07-;ACT=SGC00U<

#Create the violation report
>SED10NV4;C00+<

#Drive a LED on output 1 ON
#when the speed goes beyond the limit
#(silent event)
>SED42SV0;S07+;ACT=SSSXP11<

#Drive a LED on output 1 OFF
#when the speed falls off the limit
#(silent event)
>SED43SV0;S07-;ACT=SSSXP10<
#-----

#--START/STOP monitoring-----
```

5.8. SAFE ENGINE TURN OFF

```
#Speed limit for START/STOP detection
>SGS0310080<

#Start 2 minutes timer
>SED38SV0;S03-;ACT=SGC01TC00120<

#Stop the time
>SED39SV0;S03+ACT=SGC01U<

#Create the STOP report
>SED30NV4;C01+<

#Create the START report
>SED31NV4;S03C01&+<
#-----

#--Safe engine turn off-----
#Set the speed limit for the time/speed
#condition
>SGS0110150<

#Start 16 seconds timer when the speed
#falls off the previously set value and
#as long as the user signal is set
>SED14SV0;S01!U00&+;ACT=SGC02TC00016<

#Stop the timer when the speed condition
#is violated
>SED15SV0;S01+;ACT=SGC02U<

#Ignition cut event
#Sets output 4 high and
#sends a report to DA 5
>SED13NV5;C02U00&+;ACT=SSSXP41<

#Restore (set low) the user signal
>SED16SV0;E13+;ACT=SSSU000<
#-----

#end
```

5.8.7 Operation

Having the unit configured this way, the mechanism for driving the ignition OFF safely is:

1. To start the safe engine turn off process, activate the user signal 00.
For this send to the unit
>SSSU001<

2. If you receive a confirmation of this message the turn off process has initiated. Wait for event 13 to report. When event 13 report is received it means that the given time/speed condition met and the vehicle's engine has been turned off.
3. To restore the ignition on the vehicle simply deactivate output 4:
>SSSXP40<

The safe engine turn off process can be skipped. Although is highly unrecommended, at any time you can directly drive output 4 high making the unit turn the engine off without any previous time/speed safe condition.

Warning:

Driving a vehicle's engine OFF without knowing its state is extremely dangerous.

5.9 Improving the periodic report

The example we have been working so far uses a time-only criteria for the periodic tracking report (event 37). This report is being generated every 5 minutes 24/7. This includes a lot of useless reports that show the same location and conditions when the vehicle remains stationary. A better approach is to use all three parameters that define the *Time And Distance* signal.

Our actual definition of the T&D signal 8, which is the signal that triggers the periodic tracking event 37 is:

>STD80300<

As you can check with the TD message definition given on the TAIP reference, this is a time-only criterion which causes the TD8 signal to get high on a 5 minutes basis. To achieve a better reporting event, the T&D criterion has to include all three configuration parameters:

- Minimum time.
- Distance threshold.
- Maximum time.

These definitions depend strongly of each user's situation and it is a matter of cost vs frequency of report.

Typical values for vehicles moving in a city are:

- Minimum time between reports: 2 minutes.

5.9. IMPROVING THE PERIODIC REPORT

- Distance Threshold: 2km (1.2mi).
- Maximum time between reports: 55 minutes.

The reason for having 55 minutes instead of 1 hour, has to do with the discussion on having the unit always available for communication, presented on the *Operation* chapter.

This configuration leads to not having the unit report more frequent than a 2 minute period and having the unit to report at least once every 55 minutes. Other configurations include changing the T&D criterion as the vehicle's speed increases and/or only reporting when the vehicle's ignition is on.

To use the new T&D criterion we change our T&D 8 definition to:
>STD80120000000203555<

And leave the rest of the script unchanged.

5.9.1 Script

Changing only the TD8 definition, our script ends up:

```
#Antares SB script
#Getting Started example
# + SMS reporting (+SMS interaction + Voice interaction)
# + Ignition sense
# + Speed viol. and warning
# + START/STOP monitoring
# + Safe engine turn off
# + A better T&D criterion

#--Delete any previous configuration-----
>SRT;CONFIG<
>SXADP**U<
#-----

#--Essential configurations-----

#Units ID
>SIDEEXAMPLE<

#configuring the SIMs PIN
>SRFI1234<

#configuring the APN
>SRFAinternet.carrier-name.com<

#The remote AVL server address and port
>SXADP0000avl.server.com;2145<
```

5.9. IMPROVING THE PERIODIC REPORT

```
#The Phone-Number-type destination
#with support for user-messages ON
>SXADP1010912345678<
```

```
#A Destination Address holding the
# server destination
>SDA4;P00<
```

```
#A Destination Address holding the
# server destination and phone number
>SDA5;P00,P10<
```

```
#A complete Time And Distance criterion
#min = 120 secs.
#dist = 2km.
#max = 3555 secs.
>STD80120000000203555<
```

```
#Event triggered by T&D signal
>SED37NV4;TD8+<
```

```
#-----
```

```
#--Input sense-----
#Input report event
>SED05NV5;IP3+<
#User text message for event 05
>SXATM05Input 3 has been activated!<
#-----
```

```
#--Ignition sense-----
#Ignition ON event
>SED19NV4;F00+<
```

```
#Ignition OFF event
>SED20NV4;F00-<
#-----
```

```
#--Speed viol. + warning-----
#Set an speed limit of 65mph
>SGS0710650<
```

```
#Start a 15 secs. timer when
#speed limit is violated
#(silent event)
>SED40SV0;S07+;ACT=SGC00TC00015<
```

```
#Stop the timer when the speed falls
#below the limit
#(silent event)
```

5.9. IMPROVING THE PERIODIC REPORT

```
>SED41SV0;S07-;ACT=SGC00U<

#Create the violation report
>SED10NV4;C00+<

#Drive a LED on output 1 ON
#when the speed goes beyond the limit
#(silent event)
>SED42SV0;S07+;ACT=SSSXP11<

#Drive a LED on output 1 OFF
#when the speed falls off the limit
#(silent event)
>SED43SV0;S07-;ACT=SSSXP10<
#-----

#--START/STOP monitoring-----
#Speed limit for START/STOP detection
>SGS0310080<

#Start 2 minutes timer
>SED38SV0;S03-;ACT=SGC01TC00120<

#Stop the time
>SED39SV0;S03+ACT=SGC01U<

#Create the STOP report
>SED30NV4;C01+<

#Create the START report
>SED31NV4;S03C01&+<
#-----

#--Safe engine turn off-----
#Set the speed limit for the time/speed
#condition
>SGS0110150<

#Start 16 seconds timer when the speed
#falls off the previously set value and
#as long as the user signal is set
>SED14SV0;S01!U00&+;ACT=SGC02TC00016<

#Stop the timer when the speed condition
#is violated
>SED15SV0;S01+;ACT=SGC02U<

#Ignition cut event
#Sets output 4 high and
#sends a report to DA 5
>SED13NV5;C02U00&+;ACT=SSSXP41<
```

5.10. RECONNECTION EVENT FOR TCP

```
#Restore (set low) the user signal
>SED16SV0;E13+;ACT=SSSU000<
#-----

#end
```

5.10 Reconnection event for TCP

Depending on how the AVL server works, a TCP reconnection event may be required. As presented on the *Operation* chapter, *Antares SBTM* works as TCP client always opening a TCP connection with the server *even if it has nothing to report*. With this characteristic the unit can be interrogated at any time. The pitfall of this mechanism is that it may happen that the AVL server does not know from which unit the new incoming connection is. A solution would have been having the server to send an ID query thru the new connection so it can associate it to an unit's ID. Unfortunately not all AVL servers do this. To overcome this situation a TCP reconnection event may be configured on the unit so it sends an event every time a connection is (re) established with the server. This gives the AVL server immediate information on who has opened the connection.

A reconnection event is created simply with the **Axx** signal corresponding to the IP-type destination whose reconnection is to be monitored. So for our example the **A00** signal has to be used as the AVL server is defined on the *Destination Point* 00:

```
>SED49NV4;A00+<
```

We chose event 49 for no particular reason. This will have the AVL server receiving event 49 any time a new connection from an *Antares SBTM* is established.

5.10.1 Script

The resulting script is:

```
#Antares SB script
#Getting Started example
# + SMS reporting (+SMS interaction + Voice interaction)
# + Ignition sense
# + Speed viol. and warning
# + START/STOP monitoring
# + Safe engine turn off
# + A better T&D criterion
# + Reconnection event

#--Delete any previous configuration-----
>SRT;CONFIG<
```

5.10. RECONNECTION EVENT FOR TCP

```
>SXADP**U<
#-----

#--Essential configurations-----

#Units ID
>SIDEEXAMPLE<

#configuring the SIMs PIN
>SRFI1234<

#configuring the APN
>SRFAinternet.carrier-name.com<

#The remote AVL server address and port
>SXADP0000avl.server.com;2145<

#The Phone-Number-type destination
#with support for user-messages ON
>SXADP1010912345678<

#A Destination Address holding the
# server destination
>SDA4;P00<

#A Destination Address holding the
# server destination and phone number
>SDA5;P00,P10<

#Time-only Time And Distance
#signal definition
>STD80120000000203555<

#Event triggered by T&D signal
>SED37NV4;TD8+<
#-----

#--Input sense-----
#Input report event
>SED05NV5;IP3+<
#User text message for event 05
>SXATM05Input 3 has been activated!<
#-----

#--Ignition sense-----
#Ignition ON event
>SED19NV4;F00+<

#Ignition OFF event
```


5.10. RECONNECTION EVENT FOR TCP

```
>SED20NV4;F00-<
#-----

#--Speed viol. + warning-----
#Set an speed limit of 65mph
>SGS0710650<

#Start a 15 secs. timer when
#speed limit is violated
#(silent event)
>SED40SV0;S07+;ACT=SGC00TC00015<

#Stop the timer when the speed falls
#below the limit
#(silent event)
>SED41SV0;S07-;ACT=SGC00U<

#Create the violation report
>SED10NV4;C00+<

#Drive a LED on output 1 ON
#when the speed goes beyond the limit
#(silent event)
>SED42SV0;S07+;ACT=SSSXP11<

#Drive a LED on output 1 OFF
#when the speed falls off the limit
#(silent event)
>SED43SV0;S07-;ACT=SSSXP10<
#-----

#--START/STOP monitoring-----
#Speed limit for START/STOP detection
>SGS0310080<

#Start 2 minutes timer
>SED38SV0;S03-;ACT=SGC01TC00120<

#Stop the time
>SED39SV0;S03+ACT=SGC01U<

#Create the STOP report
>SED30NV4;C01+<

#Create the START report
>SED31NV4;S03C01&+<
#-----

#--Safe engine turn off-----
#Set the speed limit for the time/speed
```

5.11. MAIN-POWER-LOSS ALARM

```
#condition
>SGS0110150<

#Start 16 seconds timer when the speed
#falls off the previously set value and
#as long as the user signal is set
>SED14SV0;S01!U00&+;ACT=SGC02TC00016<

#Stop the timer when the speed condition
#is violated
>SED15SV0;S01+;ACT=SGC02U<

#Ignition cut event
#Sets output 4 high and
#sends a report to DA 5
>SED13NV5;C02U00&+;ACT=SSSXP41<

#Restore (set low) the user signal
>SED16SV0;E13+;ACT=SSSU000<
#-----

#--Reconnection event for DP 00-----
>SED49NV4;A00+<
#-----

#end
```

5.11 Main-power-loss alarm

If the units has a built-in back-up battery⁵, it can send a report whenever the main power source gets disconnected. To do so we use the F13 signal.

```
>SED07NV5;F13-<
```

We choose event 07 for no particular reason. Continuing with our example, this will make event 07 to be reported to the AVL server and the Phone Number 912345678 any time the main power source gets disconnected.

5.11.1 Script

Adding this message to the script is trivial so it is a task left to the reader.

5.12 Using the sleep mode

Refer to the XAPM TAIP message.

⁵ Built-in back-up battery is optional

5.13 Configuring/reading a distance counter

Refer to the XACR message.

5.14 Generating an extended-EV report

Related commands: ED, EV, XAEF

As mentioned on the *Reports' messages* section of the *Operation* chapter, an *EV* report can be extended to include extra information tags⁶. To do so, the *Message IDs A, B and C* of an *event definition* are used (See the *ED* message). An event defined with one of these letters as *Message ID* will be transmitted as an *EV* message with added information after the *Source* and *Age* fields. The extra information is presented as strings separated by the ';' character. For a detailed information on these strings refer to the *EV* message.

We are going to configure 5 events to show how the extended-EV report and the normal reports, *ET* and *EV* work.

Let us set events 20 and 21 to send *ET* and *EV* normal (not-extended) reports whenever the Counter 5 reaches some value defined elsewhere:

```
>SED20NT2;C05+<
>SED21NV2;C05+<
```

This will make the unit send the following reporting messages to *Destination Address 2* whenever signal *C05* goes from low to high:

```
>RET201447152212;ID=EXAMPLE<
>REV211447147509+2578250-0802813901519512;ID=EXAMPLE<
```

Now, let us set events 30, 31 and 32 to send extended-EV reports. Events 30 and 31 will generate an *EV* report having the extra information tags that give information on the vehicle's altitude and on the vehicle's acceleration and let us set event 32 to generate an extended report having the vehicle's acceleration, number of satellites and the state of Distance Counter 7.

First, we have to configure the extended format that it is going to be used by events 30 and 31, let us use the extended format (*XAEF*) *A*. According to the *XAEF* message the tags corresponding to altitude and acceleration are *;AL* and *;AC*. So, we set extended-format *A* as:

```
>SXAEFA;AL;AC<
```

Now, let us define extended-format *C* to be used by event 32:

```
>SXAEFC;AC;SV;CV07<
```

Having set the extended formats, define the events. For this example we are going to trigger the events with inputs:

```
>SED30NA2;IP1+<
>SED31NA2;IP2+<
>SED32NC2;IP3+<
```

⁶ See the *XAEF* message for a list of available information tags.

5.14. GENERATING AN EXTENDED-EV REPORT

Notice that events 30 and 31 use the same *Message ID* as they report the same extended information. Event 32 uses as *Message ID* the letter C. We could also have configured extended-format B instead of C.

Now, let us see the extended-EV reporting messages. Events 30 and 31 will be reported as:

```
>REV301447147509+2578250-0802813901519512;AL=00003;AC=+00;ID=EXAMPLE<
>REV311447147649+2578440-0802854301000512;AL=00001;AC=+08;ID=EXAMPLE<
```

And event 32 as

```
>REV321447147747+2578440-0802854301000512;AC=-12;CV07=01203;SV=09
;ID=EXAMPLE<
```

Notice that the order of information tags defined with the XAEF is not preserved, *Antares SBTM* reports the tags alphabetically. For detailed information see the ED, EV and XAEF messages.

6 Unit's TAIP reference

6.1. (AL) ALTITUDE

6.1 (AL) Altitude

Qualifiers: Q, R

This message gives the vehicle's altitude and vertical speed. The message has the following format:

| AAAAABBBBBBCCCCDE | | | | |
|-------------------|-------------------|--------|-------------------|---|
| Chars | Item | Format | Value | Meaning |
| 5 | GPS Time | AAAAA | seconds | GPS time of day. |
| 6 | Altitude | BBBBBB | meters | Above Mean Sea Level Altitude. It includes the + or - sign. |
| 4 | Vertical Velocity | CCCC | mph | Vertical velocity. It includes the + or - sign. |
| 1 | Data Source | D | Data Source Table | Data acquisition mode in effect when acquiring AL data. |
| 1 | Data Age | E | flag | Age of available data: 0: Not available 1: Old, 10 seconds 2: Fresh, <10 seconds |

6.2. (CP) COMPACT POSITION

6.2 (CP) Compact Position

Qualifiers: Q, R

This message gives the position solution. The message has the following format:

| AAAAABBBCCCCDDDEEEFG | | | | |
|----------------------|-------------|-----------|-------------------|---|
| Chars | Item | Format | Value | Meaning |
| 5 | GPS Time | AAAAA | seconds | GPS time of day. |
| 7 | Latitude | BBB.CCCC | degrees | Latitude in the WGS-84 datum. <i>Positive values indicate a northern latitude.</i> |
| 8 | Longitude | DDDD.EEEE | degrees | Longitude in the WGS-84 datum. <i>Positive values indicate an eastern longitude.</i> |
| 1 | Data Source | D | Data Source Table | Data acquisition mode in effect when acquiring CP data. |
| 1 | Data Age | E | flag | Age of available data: 0: Not available 1: Old, 10 seconds 2: Fresh, <10 seconds |

6.3 (DA) Destination Address

Qualifiers: Q, S, R

A *Destination Address* is an association of *Destination Points*. These allows an event defined with the ED message to be routed to multiple receivers at the same time by selecting the *Destination Address* (or group) that holds all of the desired destinations (IP-types, Telephones, Serial Port). A *Destination Address is not the actual IP address or SMS telephone of the receivers*. The message has the following format:

A;PBB[,PBB,...,PBB:PBB,...]

A is the *Destination Address*' index. Its range goes from 0 to 9. Remember that a *Destination Address* is a group or an association of *Destination Points*. A *Destination Point* (see XADP message) is the actual IP address or telephone of a destination.

B holds a *Destination Point*'s index. You can select multiple *Destination Points* by separating them with a ',' or you can select a range with a ':'.

6.3.1 Examples

Creating DAs

- To create *Destination Address* 5 as an association of *Destination Points* 2, 3, 10 and 15 (15 is the unit's serial port):
>SDA5;P02,P03,P10,P15<
- To create *Destination Address* 0 with only one *Destination Point*, for example the unit's serial port:
>SDA0;P15<
- To create a *Destination Address* as a range of DPs:
>SDA0;P00:P05,P10,P12:P13,P15<

6.4 (DP) Destination Point

Qualifiers: Q, S, R

*This command is deprecated and should not be used, use the **XADP** message instead.*

A *Destination Point* defines the IP address and/or cellular telephone of a destination. An association of *Destination Points* is used to define a *Destination Address* which is used in an *Event Definition* to route *Event Messages*. The *Destination Points* also serve security purposes as the *Antares SBTM* will only respond to queries originated from these addresses. There are 16 *Destination Points* distributed on 10 IP addresses/port, 5 Telephones and the unit's *Serial Port*. The message has the following format for destination points 00 to 09:

| AABCDDDEEEFFFGGGHHHHH | | | | |
|-----------------------|----------------------|-----------------|---------|---|
| Chars | Item | Format | Value | Meaning |
| 2 | Destination Point ID | AA | 00-09 | Decimal number holding the point's index. Indexes 00 to 09 refer to IP addresses where an AVL software is listening for TAIP messages. Indexes 10 to 14 refer to SMS receptor telephones for these indexes do not use this table, use the next one. |
| 1 | Console access | B | flag | TAIP console access flag. 1: The IP-type host has no TAIP console access. <i>Error 8 is returned on every message.</i> 0: The IP-type host has TAIP console access. U: Undefined. Delete the <i>Destination Point</i> . |
| 1 | TCP/UDP selection | C | flag | TCP/UDP and UDP-ack, UDP-no-ack selection. 0 or 1: This DP works on TCP. 2: UDP <i>without</i> confirmation. 3: UDP <i>with</i> confirmation. |
| 12 | IP address | DDD.EEE.FFF.GGG | decimal | Decimal dotted format with the receiver's numeric IP address. |
| 5 | UDP/TCP Port | HHHHH | decimal | Decimal number between 0 and 65535 with the receiver's listening UDP or TCP port |

Note: When using this message to query an IP-type destination defined with the **XADP** as a host name the IP address 0.0.0.0 is returned.

The message has the following format for destination points 10 to 14:

6.4. (DP) DESTINATION POINT

| AABC[DDD...] | | | | |
|--------------|-------------------------------|--------|-------|--|
| Chars | Item | Format | Value | Meaning |
| 2 | Destination Point ID | AA | 10-14 | Decimal number holding the point's index. Indexes 00 to 09 refer to IP addresses where an AVL software is listening for TAIP messages. Indexes 10 to 14 refer to SMS receptors' telephones. |
| 1 | Type of host or Delete action | B | | DP type/Action: U: Delete the <i>Destination Point</i> . 0: Report messages are sent as TAIP messages to this destination. 1: User-defined messages are sent instead of TAIP. See the XATM message. <i>Other values are reserved for future use.</i> |
| 1 | Access | C | | Access for this Telephone. 0: Full access. 1: TAIP console via SMS restricted, VOICE call reception allowed. 4: TAIP console via SMS allowed, VOICE call reception restricted. 5: Full restriction. |
| varies | Phone number | DDD... | Phone | Phone number of the SMS receptor. A maximum of 20 digits is allowed. |

Note:

The *Serial Port* is referenced as the *Destination Point* with index 15.

6.5 (ED) Event Definition

Qualifiers: Q, S, R

This message is used to define events. These events define the *Event Machine* configuration for the *Antares SBTM*. An event is created by defining a boolean combination of *signals* as a trigger, a routing indication for a generated *event message (EV or ET)* and a possible TAIP message to be executed when the event occurs. For more information refer to the *Event Machine* section on the *Configuration* chapter. The message has the following format:

6.5. (ED) EVENT DEFINITION

AABCD;EEE{ [EEE] [F] }G[[;ACT=HH...] [;ACT=HH...] ...]

| Chars | Item | Format | Value | Meaning |
|--------|---------------------|--------|---|--|
| 2 | Event ID | AA | 00-49 | Decimal number ranging between 00 to 49 (or '**'). |
| 1 | Event Handling | B | flag | Message routing: N: Normal. Route the <i>Event Message</i> to the specified <i>Destination Address</i> (DA). X: Serial Port. Route the <i>Event Message</i> to the unit's serial port only. S: Signal only. Do not generate an <i>Event Message</i> . The event's signal still follows the event's state. U: Undefined. Delete the event's definition. |
| 1 | Message ID | C | flag | Generate event message: V: EV message T: ET message O: Orbcomm's SC-Originated Default Message X: Binary form of the EV message A: extended-EV message A B: extended-EV message B C: extended-EV message C a: extended-EV message A for Orbcomm b: extended-EV message B for Orbcomm c: extended-EV message C for Orbcomm |
| 1 | Destination Address | D | 0-9 | Destination address of the <i>Event Message</i> . The value of this field is the index of the desired <i>Destination Address</i> (DA) defined with the DA message. |
| varies | Signal | EEE | Any signal defined on the <i>Signals' table</i> . | Signal(s) used to trigger the event. |
| 1 | Logical Operator | F | flag | Logical operation used to combine signals: &: AND : OR !: NOT |
| 1 | Event Sense | G | flag | Edge of signals' combination used to trigger the event: +: Rising edge -: Falling edge |
| varies | Event Action | HH... | message | A valid TAIP message without the opening (>) and closing (<) delimiters. Several TAIP actions can be defined on a single event. There are two valid messages to define the action. 'ACT=' which will make the event to be sent both through the serial port and over the air and 'XCT=' that will only send the event through the serial port. See the following examples. |

6.5.1 Examples

Single signal triggers

Example: *Periodic report* To define event 05 to send an EV *Event Message* every 3 minutes:

1. Use the TD message to configure a *Time & Distance* signal to trigger every 3 minutes:

```
>STD70180<
```

2. Define the event with signal TD7 as trigger:

```
>SED05NV0;TD7+<
```

Note that DA0 (*Destination Address 0*) must be defined so that the *Event Message* can be routed to any IP address(es) and/or SMS(s) receiver(s). If the message were not to be routed, the *Event Definition* message should be:

```
>SED05SV0;TD7+<
```

The only purpose of this event is to drive the E05 signal true or false according to the event's trigger (TD7+) in order to trigger any other event(s) that include E05 as part of its trigger definition.

Example: *Panic button* Send an *Event Message* when an input gets active:

```
>SED31NV0;IP3+<
```

This definition will make the unit send an EV *Event Message* with code 31 every time the input 3 becomes true.

Example: *Several actions per event* Have the unit initiate a time counter, set an user signal true and reset a distance counter whenever the vehicle's ignition goes high:

```
>SED35SV0;F00+;ACT=SGC05TC;ACT=SSSU041;ACT=SGC07U<
```

Example: *Several actions per event only sent to the serial port* Have the unit define the same actions in the previous example, but this time the actions will only be reported to the serial port:

```
>SED35SV0;F00+;XCT=SGC05TC;XCT=SSSU041;XCT=SGC07U<
```

Example: *Reporting an extended-EV message* Have the unit send an extended-EV report that includes the Altitude, Input/Outputs state and number of satellites in view whenever the input 1 is activated:

6.5. (ED) EVENT DEFINITION

>SED23NA0;IP1+;<

Define the extended-EV message *A* with the desired information tags:

>SXAEFA;AL;IO;SV;<

Example: *Reporting
another extended-EV
reporting message*

Following the previous example, create another event that generates an extended-EV report including the state of distance counter 12 whenever the vehicle is turned off:

>SED24NB0;F00-;<

>SXAEFB;CV12;<

*Binary form of the EV
message*

The binary form of Antares' EV messages is configured by simply using "X" as the Message ID qualifier in the event definition (ED). It is important to note that the Destination Address of the messages must contain an IP-type Destination Point and that the destination server must be configured to decode the binary messages.

Application notes AN0020EN and AN0021EN explains the binary form and a method to decode the binary messages sent by *Antares SBTM* that can be implemented in any server. Both application notes can be found at: <http://www.digitalcomtech.com/support.html>

Example: *Send an Event
Message in binary form*

Send an *Event Message* in binary form when an input gets active:

>SED41NX0;IP4<

This definition will make the unit send an EV *Event Message* in binary form with code 41 every time the input 4 becomes true. For this example it is assumed that the DA0 contains a valid IP-type *Destination Point*. A regular TAIP message will be sent to any *Destination Point* that is not an IP-type *Destination Point* if they are included in the DA.

If DA0 does not contain a *Destination Point* with a valid IP-type destination or contains other type of *Destination Point* as well as the IP-type *Destination Point* the message will be send as regular TAIP messages.

Refer to the *Scenarios and examples* chapter for more information.

6.6 (ER) Error Report

Qualifiers: R

This message is generated by the unit whenever the previous command entered on the TAIP console generates an error. The message has the following format:

| AA: [BBB...] | | | | |
|--------------|--------------|--------|-------|--|
| Chars | Item | Format | Value | Meaning |
| 2 | Error Code | AA | | Error code: See the table of error codes at the end of this chapter. |
| varies | TAIP message | BBB... | | TAIP message that generated the error. It does not include the message's '>' and '<' characters. |

6.6.1 Example

Entering the following command will make the unit generate error 86:
>SGF00<

The unit returns:
>RER86:SGF00<

That according to the error list:

Can not change a GPIOs mask (GF) on this product. Only the F0 value is accepted. See the GF message.

6.7. (ET) EVENT REPORT, TIME ONLY MESSAGE

6.7 (ET) Event Report, time only message

Qualifiers: R

This message is generated when an event is triggered. This message only contains date and time of the generated event. The message has the following format:

| AABBBBCDDDDD | | | | |
|--------------|-----------------|--------|---------|---|
| Chars | Item | Format | Value | Meaning |
| 2 | Event code | AA | 00-49 | Event code. |
| 4 | GPS Week | BBBB | week | Number of weeks since 00:00AM January 6, 1980. The event's date can be deduced from this number. |
| 1 | Day of week | C | day | Day of week. From 0 to 6 where 0 is Sunday. |
| 5 | GPS Time of day | DDDDD | seconds | Time of the generated report. |

6.8. (EV) EVENT MESSAGE

6.8 (EV) Event Message

Qualifiers: R

This message is generated when an event is triggered and reported. The message has the following format:

AABBBBCDDDDDEEEFFFFFGGGGHHHHHIIJJJKL[EXTENDED-EV TAGS]

| Chars | Item | Format | Value | Meaning |
|-------|------------------|---------------|---------|---|
| 2 | Event code | AA | 00-49 | Event code. |
| 4 | GPS Week | BBBB | week | Number of weeks since 00:00AM January 6, 1980. The event's date is deduced from this number. |
| 1 | Day of week | C | day | Day of week. From 0 to 6 where 0 is Sunday. |
| 5 | GPS Time of day | DDDDD | seconds | Time of the generated report. |
| 8 | Latitude | EEE.FFFFF | degrees | WGS-84 Latitude. It does include the sign: <i>Positive for north</i> . |
| 9 | Longitude | GGGG.HHHHH | degrees | WGS-84 Longitude. It does include the sign: <i>Positive for east</i> . |
| 3 | Speed | III | mph | Vehicle velocity. |
| 3 | Heading | JJJ | degrees | Vehicle heading, in degrees from North increasing eastwardly. |
| 1 | Source | K | flag | Position fix mode: 0: 2D GPS 1: 3D GPS 2: 2D DGPS 3: 3D DGPS 6: DR 8: Degraded DR 9: Unknown |
| 1 | Age of data | L | flag | Age of data used for the report: 0: Not available 1: Old, 10 seconds 2: Fresh, <10 seconds |
| | Extended-EV Tags | [;xxx;xxx...] | | Information Tags: See the following table. |

6.8. (EV) EVENT MESSAGE

| Tag | Meaning | | | | | | | | | | | | | | | | | | | | | | |
|---------------------|---|------|---------------------------------|------|---|------|--|------|---------------------------------|------|----------|------|----------|------|----------|------|--------------------------------|------|---------|------|---------|------|---------|
| ;AL=AAAAA | Altitude is A meters Above Mean Sea Level (AMSL). | | | | | | | | | | | | | | | | | | | | | | |
| ;AC=AAA | Acceleration is A [Miles] per [Hour per second] ^a . | | | | | | | | | | | | | | | | | | | | | | |
| ;AD=AAAAA | Voltage on the Analog port is A mV. | | | | | | | | | | | | | | | | | | | | | | |
| ;BL=AAAAA | Voltage of the back-up battery is A mV. | | | | | | | | | | | | | | | | | | | | | | |
| ;CE=AAAABB | Simple Cell ID information. (See the XACE message for information on each item) A: Cell ID in hex. B: RxLvl. | | | | | | | | | | | | | | | | | | | | | | |
| ;CF=AAABBBCCCCDDDEE | Full Cell ID information. (See the XACE message for information on each item) A: MCC. Mobile Country Code B: MNC. Mobile Network Code C: LAC (hex). Local Area Code D: Cell ID (hex). E: RxLvl. | | | | | | | | | | | | | | | | | | | | | | |
| ;CVAA=BBBBB | The value of counter A is B. | | | | | | | | | | | | | | | | | | | | | | |
| ;IO=ABC | Inputs and Outputs state. Bitwise representation of A: <table border="1"> <tr> <td>bit0</td><td>Ignition (1=ACTIVE, 0=INACTIVE)</td></tr> <tr> <td>bit1</td><td>Main Power Source (1=EXT-PWR, 0=BACKUP-BATTERY)</td></tr> <tr> <td>bit2</td><td>12/24V detector. (1=EXT-PWR-AT-24V 0=EXT-PWR-AT-12V)</td></tr> </table> Bitwise representation of B (outputs): <table border="1"> <tr> <td>bit0</td><td>Output 1 (1=ACTIVE, 0=INACTIVE)</td></tr> <tr> <td>bit1</td><td>Output 2</td></tr> <tr> <td>bit2</td><td>Output 3</td></tr> <tr> <td>bit3</td><td>Output 4</td></tr> </table> Bitwise representation of C (inputs): <table border="1"> <tr> <td>bit0</td><td>Input 1 (1=ACTIVE, 0=INACTIVE)</td></tr> <tr> <td>bit1</td><td>Input 2</td></tr> <tr> <td>bit2</td><td>Input 3</td></tr> <tr> <td>bit3</td><td>Input 4</td></tr> </table> | bit0 | Ignition (1=ACTIVE, 0=INACTIVE) | bit1 | Main Power Source (1=EXT-PWR, 0=BACKUP-BATTERY) | bit2 | 12/24V detector. (1=EXT-PWR-AT-24V 0=EXT-PWR-AT-12V) | bit0 | Output 1 (1=ACTIVE, 0=INACTIVE) | bit1 | Output 2 | bit2 | Output 3 | bit3 | Output 4 | bit0 | Input 1 (1=ACTIVE, 0=INACTIVE) | bit1 | Input 2 | bit2 | Input 3 | bit3 | Input 4 |
| bit0 | Ignition (1=ACTIVE, 0=INACTIVE) | | | | | | | | | | | | | | | | | | | | | | |
| bit1 | Main Power Source (1=EXT-PWR, 0=BACKUP-BATTERY) | | | | | | | | | | | | | | | | | | | | | | |
| bit2 | 12/24V detector. (1=EXT-PWR-AT-24V 0=EXT-PWR-AT-12V) | | | | | | | | | | | | | | | | | | | | | | |
| bit0 | Output 1 (1=ACTIVE, 0=INACTIVE) | | | | | | | | | | | | | | | | | | | | | | |
| bit1 | Output 2 | | | | | | | | | | | | | | | | | | | | | | |
| bit2 | Output 3 | | | | | | | | | | | | | | | | | | | | | | |
| bit3 | Output 4 | | | | | | | | | | | | | | | | | | | | | | |
| bit0 | Input 1 (1=ACTIVE, 0=INACTIVE) | | | | | | | | | | | | | | | | | | | | | | |
| bit1 | Input 2 | | | | | | | | | | | | | | | | | | | | | | |
| bit2 | Input 3 | | | | | | | | | | | | | | | | | | | | | | |
| bit3 | Input 4 | | | | | | | | | | | | | | | | | | | | | | |

^a See the *Configuration* chapter for information on how vehicle's acceleration is measured.

6.8. (EV) EVENT MESSAGE

| Tag | Meaning |
|------------|---|
| ;NS=ABCCDE | GSM/GPRS Network Status: A Sim Card State: Same as the <i>SIM state</i> field of the XADM message. B GSM Registration Status: Same as the <i>GSM Status</i> field of the XADM message. C Received signal strength: Same as the <i>RSSI</i> field of the RP message. D GPRS 'Attached' flag E GPRS session state: Same as the <i>GPRS state</i> field of the XADM message. |
| ;SV=AA | A GPS satellites in view. |
| ;OE=AAAAA | Instantaneous Engine's RPM (OBD Related). |
| ;OT=AAA | Instantaneous Throttle, 0 100% (OBD Related). |
| ;OF=AAA | Fuel Level, 0 100% (OBD Related). |
| ;OG=AAAAA | Remaining fuel gallons (OBD Related). |
| ;OR=AAA | Instantaneous Fuel Rate, 000 to 029 gallons per hour (OBD Related). |
| ;OS=AA | OBD Status. This is a HEX value that must be converted to Binary where each bit represents a flag. The flags from the Least Significant Bit to the Most Significant Bit are: <ul style="list-style-type: none"> •<i>SignalState</i> True if the OBD is communicating with <i>Antares SBTM</i>. •<i>ChecksumError</i> True if a checksum error was detected. •<i>Ignition</i> True if the vehicle's engine is on. •<i>MIL</i> True if the MIL (Malfunction Indicator Light) is on. •<i>SecondaryTool</i> True if a secondary OBD tool is connected. |
| ;OD=AAA... | Vehicle's Odometer in miles, 1 to 10 characters long (OBD Related). |
| ;OI=AAA... | Trip Odometer in .1 miles, 1 to 10 characters long (OBD Related). |
| ;VO=AAA... | Virtual Odometer value, 1 to 10 characters long. |
| ;RE=ABCC | Region Event. A I: Unit entered region. O: Unit left region. B Type of Region: K: Circular region. R: Polygonal region. CC Region Index. |

The extended-EV messages are generated with events that use as *Message ID* one of the extended-EV formats A, B or C. For more information refer to the *ED* and *XAEF* messages.

6.9 (GC) Counters, Timers, Distancers

Qualifiers: Q, S, R

This message is used to configure and manipulate internal counters. Each counter can be configured as a user-controlled counter, a timer, or a distancer (counter updated by the traveled distance). For more information refer to the *Configuration* chapter, *Using Counters* section. The message has the following format:

| AAB[C[DDDDD[EEEE]]] | | | | |
|---------------------|--------------|--------|------------------------------|---|
| Chars | Item | Format | Value | Meaning |
| 2 | Counter ID | AA | 00-19 | Specifies one of the counters that it is associated with a counter signal CAA. The counter signal becomes "True" when the specified threshold value is reached. If the Recycle Flag is set to "R" then the signal transitions back to "False". If the Recycle Flag is set to "C" the signal will remain "True". |
| 1 | Command | B | See Counter's Commands table | A counter can be configured in one of three modes using the Counter (C), Timer (T), or Distancer (D) command. |
| 1 | Recycle Flag | C | flag | Action performed when the counter threshold is reached: R: Recycle counter (set to zero) C: Continue counter X: Use existing or default Recycle Flag |
| 5 | Value | DDDDD | seconds | Threshold counter increment when used with the I command or set with the V command. |
| 5 | Value | EEEE | seconds | Data increment value for Counter modes. For Timers, the counter value is incremented by 1 for every Delta elapsed seconds. For Distancers, the counter value is incremented by 1 for every Delta accumulated meters traveled. |

6.9.1 Counters' commands

| Value | Meaning |
|-------|--|
| C | Sets the counter type to <i>Counter</i> with a threshold value defined as DDDDD. The value for this type of counter can only be changed with the <i>increment</i> (I or <i>value</i> (V) command. |
| T | Set and start a <i>Timer</i> counter with a threshold value defined as DDDDD, time increment 1 or EEEEE seconds. |
| D | Set and start a <i>Distance</i> counter with a threshold value defined as DDDDD, distance increment 1 or EEEEE meters. |
| S | Counter update is suspended. |
| R | Resume a suspended counter. The time or distance increments occurring while the counter were suspended are not added upon resume. |
| I | Only for <i>Counters-type</i> counters: Increment the counter's value by 1 or DDDDD. |
| V | When using the S qualifier: Set the counter's value to DDDDD. Use the Q qualifier to get the actual counter's value. |
| U | Undefine counter: The counter's definitions is deleted and the associated counter signal CAA is reset. |

6.9.2 Examples

Example 1 Set counter 03 on *Timer* mode. When the counter's value reaches 5 minutes the C03 signal should get true. The counter shall not recycle its value when reaching the 5 minutes or else we will end up with a periodic C03 signal:

To do this we define a timer with threshold value set to 300 seconds with no delta value:

```
>SGC03TC00300<
```

Notice the *Recycle Flag* set to C so the counter does not resets when reaching the threshold. Now 5 minutes after entering this command we will have the C03 signal transitioning from false to true.

Example 2 Use a timer to generate a periodic counter signal having a period of 27 minutes:

To do this we define a timer that recycles whenever the count value reaches the threshold. To show the use of the delta parameter we are not going to count seconds but minutes:

```
>SGC07TR000270006Q<
```

Now C07 signal gets high every 27 minutes (getting low immediately, after all event's are evaluated).

Example 3 To create a distance counter that count x100 meters with no particular interest on any counter signal:

```
>SGC05DC0000100100<
```

We set the threshold to 1 but it could have been set to any value as we

6.9. (GC) COUNTERS, TIMERS, DISTANCERS

are not interested on monitoring the C05 signal.

We can query the distance counter value at any time:

>QGC05V<

Ant the unit shall respond:

>RGC05V00997<

Indicating that the traveled distance since the counter set is 99.7km (997 x 100m).

Example 4 We can also consult the actual settings of a counter:

>QGC05<

The unit responds with the actual distance counter settings:

>RGC05DC0000100100<

6.10 (GF) GPIOs' function (I/O)

Qualifiers: Q, S, R

This message is deprecated and it should not be used. If used, only the value F0 is accepted.

This message is used to configure the unit's GPIOs Input/Output function.

Note:

Antares SBTM does not have GPIOs. It has 4 inputs and 4 outputs instead.

The message has the following format:

| AA | | | | |
|-------|-----------------|--------|-------------|--|
| Chars | Item | Format | Value | Meaning |
| 2 | GPIOs' I/O Mask | AA | Hexadecimal | The GPIOs input/output mask. An <i>output</i> is defined by setting the bit that corresponds to the GPIO's index. An <i>input</i> is defined by re-setting this bit. |

Only the following command works on *Antares SBTM*:

>SGFF0<

Any other value will make the unit return error 86:

>SGF00<

>RER86:SGF00<

6.11 (GS) Speed Limit

Qualifiers: Q, S, R This message is used to configure the speed limits that can be used to trigger events. The message has the following format:

| AABCCCC | | | | |
|---------|----------------|--------|--------|---|
| Chars | Item | Format | Value | Meaning |
| 2 | Speed limit ID | AA | 00-09 | Identification code assigned to speed limit. |
| 1 | Active flag | B | flag | 1: Speed limit is active. U: Delete speed limit. |
| 4 | Speed limit | CCCC | 0-9999 | Speed limit in miles per hour times 10. |

See the *Setting Speed limits* section on the *Configuration* chapter for examples.

6.12 (GT) Time Window

Qualifiers: Q, S, R

This message is used to configure the *Time Windows* that can be used to trigger events. The signal associated with a *Time Window* becomes active when the actual date and time is between the values set by the **TMin** and **TMax** parameters. Note: The minimum time between TMin and TMax is 2 minutes. The message has the following format:

| AAB[CCCCCDDDDDEEEEEFFFFF] | | | | |
|---------------------------|----------------|----------|----------|---|
| Chars | Item | Format | Value | Meaning |
| 2 | Time window ID | AA | 00-09 | Identification code assigned to time window. |
| 1 | Active flag | B | flag | 1: Time window is active. U: Delete Time window. |
| 6 | TMin[Day] | CC/CC/CC | yy/mm/dd | Date at beginning of time window. |
| 6 | TMin[Time] | DD:DD:DD | hh:mm:ss | Time at beginning of time window. |
| 6 | TMax[Day] | EE/EE/EE | yy/mm/dd | Date at end of time window. |
| 6 | TMax[Time] | FF:FF:FF | hh:mm:ss | Time at end of time window. |

If a day is not specified (TMin[Day] and TMax[Day] are both 000000), TMin[Time] and TMax[Time] specify a periodic window.

6.13 (GR) Regions

Qualifiers: Q, S, R

Use this message to create/edit polygon-defined regions. Managing a region is different than managing any other configuration parameter, this because a region can't be defined only with one **GR** message, instead, several messages are used to create a region, one for every point the region has. So when you use a **GR** message, you are creating or editing only a region's point.

The *Active Flag* of the message is used to set the last point of a region. To undefine a region simply set the *Active Flag* of the first or second point¹ to U. The message has the following format:

| AABBCDDDEEEEEFFFFGGGGHHII | | | | |
|---------------------------|-------------------------|------------|----------|---|
| Chars | Item | Format | Value | Meaning |
| 2 | Region index | AA | 0-29 | Region index of the point that is being configured. |
| 2 | Point index | BB | 00-49 | Point index. |
| 1 | Last point flag | C | flag | Use: 1: The actual point is not the last point of the region. U: The actual point is the last point of the region. Use this flag on points 00 or 01 to undefine a region. |
| 8 | Point's latitude | DDD.EEEEE | degrees | WGS-84 point's latitude. It does include sign: <i>Positive for north</i> . |
| 9 | Point's longitude | FFFF.GGGGG | degrees | WGS-84 point's longitude. It does include sign: <i>Positive for east</i> . |
| 2 | Point's left neighbor 1 | HH | 00-49,99 | Index of the first <i>left neighbor</i> of the actual point. Use 99 to leave this field empty. <i>Check the explanation after this table.</i> |
| 2 | Point's left neighbor 2 | II | 00-49,99 | Index of the second <i>left neighbor</i> of the actual point. Use 99 to leave this field empty. <i>Check the explanation after this table.</i> |

The *Antares SBTM* has the ability to detect whether a vehicle is inside or outside of a user defined polygon. In order to achieve this the point's numeration must follow the next rule:

The points must be numerated (starting on 0 and going up to a maximum of 49) in ascending order according to each point's longitude. If two or more points share the same longitude their index ordering is not relevant.

If the polygon were drawn with the geographical north pointing upwards this rule will be achieved by numerating the points from left to right.

¹ A polygon is defined by a minimum of three points

6.13. (GR) REGIONS

A point is formed by the *region's index*, the *point's index*, the *point's latitude and longitude* and by other two parameters called the *left neighbors*. The *left neighbors* of a point are the indexes of the points connected (if any) with a line to the actual point and have a longitude value *equal to or less than* the actual point's longitude. When the actual point has no left neighbors (there are no connected points with the same or less longitude) these two parameters take the value of 99 each, as there is no point with 99 as index².

6.13.1 Special cases

- When the actual point has only one *left neighbor* the first parameter takes the neighbor's index and the second must be 99.
- When two points lie on the same longitude it is not relevant which is the point with lower index, but a special consideration must be taken:

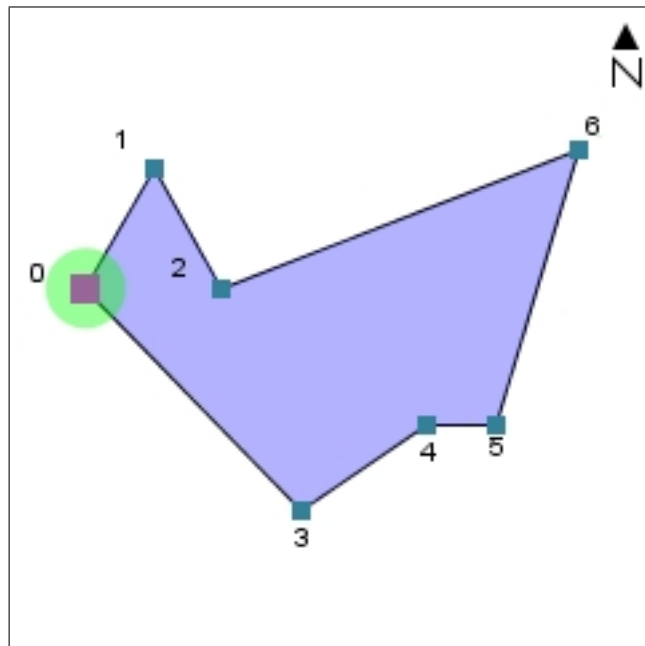
When two connected points lie on the same longitude only one of them, whichever is selected, is left neighbor of the other.

- In case that a region has to cross the longitudes E179.99999 or W179.99999 or the latitudes N89.99999 or S89.99999 the region must be divided in such a way that it never crosses those boundaries.

6.13.2 Regions' creation examples

Take a look to the following regions' examples that follow the directions discussed above.

Example: *Region A* This is a region where there are not special cases. The region has 7 points.



² A maximum of 50 points may be defined for a single region

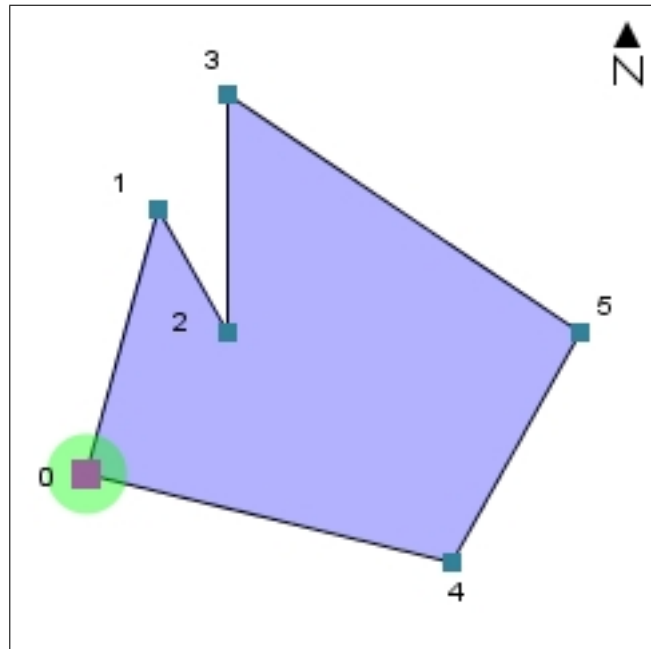
6.13. (GR) REGIONS

This region is drawn with the north pointing upwards so you can see how the leftmost point has the index 00 and the rightmost has the higher index 06 following the longitude ordering discussed above.

Left neighbors Lets examine each point's *left neighbors* according to the definition given.

- Point 00 has no points on the left, so it has no *left neighbors*. These parameters should be filled with 99 each.
Left Neighbor 1 : 99
Left Neighbor 2 : 99
- Point 01 has point 00 to its left and is connected to it, so this is a *left neighbor*. There are no more points to the left of 01, so the second neighbor parameter must be 99.
Left Neighbor 1 : 00
Left Neighbor 2 : 99
- Point 02 has points 00 and 01 to its left but only point 01 is connected to it, so this its only *left neighbor*.
Left Neighbor 1 : 01
Left Neighbor 2 : 99
- Point 03.
Left Neighbor 1 : 00
Left Neighbor 2 : 99
- Point 04.
Left Neighbor 1 : 03
Left Neighbor 2 : 99
- Point 05.
Left Neighbor 1 : 04
Left Neighbor 2 : 99
- Point 06 has two *left neighbors*.
Left Neighbor 1 : 02
Left Neighbor 2 : 05

Example: *Region B* This a region where two connected points share the same longitude.



As you can see points 02 and 03 share the same longitude. As discussed early it is not relevant which point comes first, the points could be assigned having the 03 below the 02 instead of the arrangement presented in the figure. The relevant issue is that these two points are connected and you must be careful of counting only one connection. Lets take a look to the *left neighbors* to clarify this.

Left neighbors

- Point 00 has no points on the left, so it has no *left neighbors*.
 Left Neighbor 1 : 99
 Left Neighbor 2 : 99
- Point 01.
 Left Neighbor 1 : 00
 Left Neighbor 2 : 99
- Point 02 has point 01 as *left neighbor*. Now we can decide to add point 03 as the second *left neighbor* but we must be careful of not counting 02 as *left neighbor* of point 03.
 Left Neighbor 1 : 01
 Left Neighbor 2 : 03
- Point 03 has no *left neighbors* because the only point connected to it is 02 but it has the same longitude and we already counted this connection on the previous point. This leaves us with
 Left Neighbor 1 : 99
 Left Neighbor 2 : 99

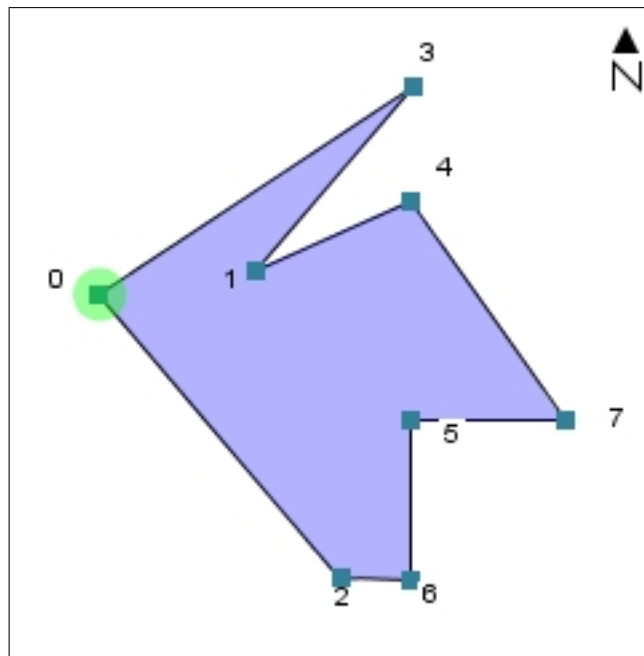
6.13. (GR) REGIONS

Note:

Another approach would have been counting 02 as *left neighbor* of point 03 but then we must be careful of not including 03 as *left neighbor* of point 02.

- Point 04.
Left Neighbor 1 : 00
Left Neighbor 2 : 99
- Point 05.
Left Neighbor 1 : 03
Left Neighbor 2 : 04

Example: *Region C* A general example.



Left neighbors

- Point 00.
Left Neighbor 1 : 99
Left Neighbor 2 : 99
- Point 01.
Left Neighbor 1 : 99
Left Neighbor 2 : 99
- Point 02.
Left Neighbor 1 : 00
Left Neighbor 2 : 99

6.13. (GR) REGIONS

- Point 03.
Left Neighbor 1 : 00
Left Neighbor 2 : 01
- Point 04.
Left Neighbor 1 : 01
Left Neighbor 2 : 99
- Point 05.
Left Neighbor 1 : 06
Left Neighbor 2 : 99
- Point 06.
Left Neighbor 1 : 02
Left Neighbor 2 : 99
- Point 07 has two *left neighbors* that share the same longitude. Only in this case you can choose either point as *left neighbor 1* and the other as *left neighbor 2*.
Left Neighbor 1 : 04
Left Neighbor 2 : 05

6.14 (ID) Identification

Qualifiers: Q, S, R

This message is used to set/query unit's ID. The message has the following format:

| A [AAA . . .] | | | | |
|---------------|------------|--------------|--------|---|
| Chars | Item | Format | Value | Meaning |
| 1 to 10 | Vehicle ID | A[AAA . . .] | string | Identification code assigned to the vehicle. This parameter may be alpha-numeric. The only forbidden characters are <, > and ; . The factory default is 0000. |

6.15 (MS) Memory Session

Qualifiers: S, R

This message is deprecated and has no effect on the unit.

6.16 (MT) MDT Mode

Qualifiers: Q, S, R

This message controls and starts the serial port's MDT communication mode. The message has the following format:

A[BBB[CCCD\EE\FF\GG\HH]] [;ROUTE=I]

| Chars | Item | Format | Value | Meaning |
|-------|---------------------------|--------|-----------|---|
| 1 | Mode | A | flag | MDT mode: N: Normal(default) P: PAD R: ORBCOMM satellite modems O: OBD Mode H: Full Binary |
| 3 | Timeout | BBB | seconds | Packet timeout. (PAD mode) |
| 3 | Max. Length | CCC | | Packet max. size (PAD mode). Default is 50 (max. is 50). |
| 1 | Include packet delimiter | D | flag | Include packet delimiter (PAD mode): T: Include F: Exclude |
| 3 | Packet delimiter 1 | \EE | hex value | Code for the character that is to be used as delimiter 1 (PAD mode). Default is \0D. |
| 3 | Packet delimiter 2 | \FF | hex value | Code for the character that is to be used as delimiter 2 (PAD mode). Default is \0A. |
| 3 | PAD mode escape character | \GG | hex value | Code for the character that is to be used to end PAD mode and go back to NORMAL mode. Default is \1B (escape character). |
| 3 | reserved | | | Must be set to \FF. |
| 1 | Routing option | I | 0-F | <i>Destination Point</i> index to send the resulting TX messages. One character corresponding to the hex value of the <i>DP</i> (i.e. 00-15 = 0-F). |

For more information and examples on the MDT PAD mode see the *Serial Port Devices* section of the *Configuration* chapter. Also, refer to the *ORBCOMM Satellite Modems Support* and the *OBD Support* sections of the *Operation* chapter for more information on these modes. The TX message is also related to this mode.

6.17 (PV) Position-velocity*Qualifiers:* Q, R

This message gives the unit's current position, velocity, heading, source of information and age of the data. The message has the following format:

| AAAAABBBCCCCDDDEEEEEFFFGGGHI | | | |
|------------------------------|------------|---------|---|
| Item | Format | Units | Meaning |
| GPS Time of day | AAAAA | seconds | Time of the generated report. |
| Latitude | BBB.CCCCC | degrees | WGS-84 Latitude. It does include the sign: <i>Positive for north.</i> |
| Longitude | DDDD.EEEEE | degrees | WGS-84 Longitude. It does include the sign: <i>Positive for east.</i> |
| Speed | FFF | mph | Vehicle velocity. |
| Heading | GGG | degrees | Vehicle heading, in degrees from North increasing eastwardly. |
| Source | H | flag | Position fix mode: 0: 2D GPS 1: 3D GPS 2: 2D DGPS 3: 3D DGPS 6: DR 8: Degraded DR 9: Unknown |
| Age of data | I | flag | Age of data used for the report: 0: Not available 1: Old, 10 seconds 2: Fresh, <10 seconds |

6.18 (RF) Radio Frequency module configuration

Qualifiers: Q, S, R

This message is used to configure Cellular Network parameters.

Note:

Any RF parameter can be left empty by issuing the command without the B string

The message has the following format:

| A [BBB . . .] | | | |
|---------------|--------|---------------|---|
| Item | Format | Value | Meaning |
| Parameter ID | A | <i>char</i> | Values: I: SIM Card PIN. A: GPRS APN (Access Point Name) (40 chars. max.). L: GPRS Login (40 chars. max.) P: GPRS Password (40 chars. max.) |
| Parameter | B | <i>string</i> | String with the parameter described by A. It can not contain the < or the ; characters. |

See the *Configuration* chapter for examples.

6.19 (RM) Reporting Mode

Qualifiers: Q, S, R

This message controls flags used to change the format of the responses and reports, i.e. messages with the **R** qualifier. The message has the following format:

| [;ID_FLAG=A] [;EC_FLAG=B] [;CR_FLAG=C] [;ER_FLAG=D] [;CS_FLAG=E] | | | | |
|--|---------|--------|-------|--|
| Chars | Item | Format | Value | Meaning |
| 1 | ID Flag | A | flag | Include the unit's ID on each message with a ";ID=" postfix. T: True (Default). F: False. |
| 1 | EC Flag | B | flag | When set, the unit will echo any correct message that uses the S qualifier. T: True (Default). F: False. |
| 1 | CR Flag | C | flag | When set, a CR and LF is appended to every unit's response/report. T: True (Default). F: False. |
| 1 | ER Flag | D | flag | When set, enables error responses. T: True (Default). F: False. |
| 1 | CS Flag | E | flag | When set, enables checksum responses. T: True. F: False (Default). |

6.20 (RP) Registration Parameters (Cellular Network)

Qualifiers: Q, R

This message reports the unit's cellular registration state. The message has the following format:

| ABBCCCC | | | |
|--------------------|--------|---------------------|--|
| Item | Format | Units | Meaning |
| Registration state | A | 0-E | <p>Values:</p> <ul style="list-style-type: none"> 0: Not registered and not searching operator. 1: Registered in home network. 2: Not registered and searching operator. 3: Not registered. Registration denied. 4: Unknown. This state should be temporal. 5: Registered and roaming. 6: Not Registered. Network lock. 7: Not Registered. SIM PIN error. 8: Not Registered. SIM PIN wait. 9: Not Registered. SIM inserted. A: Not Registered. SIM removed. B: Not Registered. SIM PIN OK. C: Not Registered. Error reading PIN. D: Not Registered. SIM initialization not done yet. E: Not Registered. Unknown SIM error. |
| RSSI | BB | 00-31 | RSSI: Received Signal Strength Indicator. This a decimal value that indicates the power of the received signal. See explanation at the end of the section. |
| RSS | CCCC | dBm -113 to -051 | RSS: Received Signal Strength. Indicates the power of the received signal. See explanation at the end of the section. |

Check the following table to see the relation between RSSI and RSS.

6.20. (RP) REGISTRATION PARAMETERS (CELLULAR NETWORK)

| RSSI | RSS _{GSM} [-dBm] | |
|----------|---------------------------|-----------|
| 0 to 7 | 113 - 99 | Very poor |
| 8 to 13 | 97 - 87 | Poor |
| 14 to 17 | 85 - 79 | Fair |
| 18 to 20 | 77 - 73 | Fair |
| 21 to 24 | 71 - 65 | Good |
| 25 to 31 | 63 - 51 | Excellent |

6.21 (RT) Reset message

Qualifiers: S, R

When this message is issued alone the unit performs a system reset. When used with qualifiers it serves multiple internal initializations purposes. The message has the following format:

| [;CONFIG] [;SIGNALS] | |
|----------------------|--|
| Qualifier | Meaning |
| ;SFBUFF | Deletes the contents of the S&F buffer. |
| ;CONFIG | Resets almost all the configuration of the unit. The preserved parameters are: PIN, APN, ID, <i>Destination Points</i> , IMEI as ID. This prevents losing communication over the air with the unit. In order to delete absolutely all off the unit's parameters use the ;ALL option. |
| ;ALL | Resets all of the configuration of the unit. This option can not be used over the air. |

6.22 (SS) Signal Status

Qualifiers: Q, S, R

This message allows for the inspection of signals' state and the *setting of outputs and other manipulatable signals*. See the Signals' list on the *Event Machine* section of the *Configuration* chapter. The message has the following format:

| AAA[B] | | | | |
|--------|-----------------------|--------|--------------------------|---------------------------------------|
| Chars | Item | Format | Value | Meaning |
| 3 | Signal ID | AAA | See <i>Signals'</i> list | 3-character identifier of the signal. |
| 1 | Current signal status | B | 0 or 1 | Status of the specified signal. |

6.22.1 Examples

Asking To know the state of the Region signal 03:

>QSSR03<

If the signal is OFF (the vehicle is not inside region 03) the unit returns:

>RSSR030<

If the signal is ON (the vehicle is inside region 03):

>RSSR031<

Asking Inputs To know the state of the input 2:

>QSSIP2<

If input 2 is inactive the unit returns:

>RSSIP20<

If input 2 is active the unit returns:

>RSSIP21<

Setting To set the user signal 08 to false:

>SSSU080<

To set it true:

>SSSU081<

Setting outputs To activate output 3

6.22. (SS) SIGNAL STATUS

>SSSXP31<

To deactivate it:

>SSSXP30<

See the *Configuration* and *Operation* chapters for more examples.

6.23. (ST) STATUS

6.23 (ST) Status

Qualifiers: Q, R

This message provides information about the unit's GPS receiver. The message has the following format:

| AABCDDEFGG | | | | |
|------------|------------------------|--------|-----------------------|--|
| Chars | Item | Format | Value | Meaning |
| 2 | Tracking Status Code | AA | Hex characters: 00-0A | GPS satellite signal acquisition and tracking status. 00: Doing position fixes. 01: Don't have GPS time yet. 02: Not used. 03: PDOP is too high. 08: No usable satellites. 09: Only 1 usable satellite. 0A: Only 2 usable satellites. 0B: Only 3 usable satellites. BB: Stationary Mode. 0C: Chosen satellite is unusable. |
| 1 | Status Codes, Nibble 1 | B | Hex characters: 0-F | Value: 0: No problems reported. 2: Antenna feedline open fault. 6: Antenna feedline short fault. |
| 1 | Status Codes, Nibble 2 | C | Hex characters: 0-F | Value: 0: No problems reported. 1: Battery back up failed; RAM not available at power-up (see <i>note</i> below). |
| 2 | Machine ID | DD | Hex characters: 00-FF | Internal GPS Machine ID |
| 1 | Status Codes, Nibble 3 | E | Hex characters: 0-F | Not used. |
| 1 | Status Codes, Nibble 4 | F | Hex characters: 0-F | Value: 0: No problems reported. 2: RTC not available at power-up (see <i>note</i> below). 8: Stored almanac not complete and current. A: RTC not available, stored almanac not complete and current. |
| 2 | Not used | GG | Hex value | Not used. |

Note - After the status is detected, this bit remains set until the receiver is reset.

6.24. (TM) TIME AND DATE

6.24 (TM) Time and Date

Qualifiers: Q, R

This message provides information about the unit's Time and Date as provided by the GPS service. The message has the following format:

| AABBCCDDDEEFFGGGGHHIJJKLLLL | | | | |
|-----------------------------|-----------------------------|--------|----------------|---|
| Chars | Item | Format | Value | Meaning |
| 2 | Hour | AA | <i>decimal</i> | |
| 2 | Minutes | BB | <i>decimal</i> | |
| 5 | Seconds | CC.DDD | | |
| 2 | Day | EE | <i>decimal</i> | |
| 2 | Month | FF | <i>decimal</i> | |
| 4 | Year | GGGG | <i>decimal</i> | |
| 2 | GPS UTC Time Offset | HH | seconds | |
| 1 | Fix Mode | F | | Value: 0: 2D GPS 1: 3D GPS 2: 2D DGPS 3: 3D DGPS 6: Reserved 8: Reserved 9: No fix available |
| 2 | Number of usable satellites | JJ | | |
| 1 | GPS UTC Offset flag | K | Flag | 0: Invalid. 1: Valid. |
| 5 | Reserved | LLLLL | | |

6.25 (TD) Time and Distance signals configuration

Qualifiers: Q, S, R

The *Time and Distance* signals are set by its corresponding *Time and Distance* counter which is a counter that follows a *Time and Distance* criteria. This criteria allows to create a counter that does not follow a time or distance criteria independently from each other, instead, combines these two variables to generate an intelligent trigger to be used for a more efficient vehicle tracking. *These signals are immediately reset by the processor after being evaluated in order to allow future triggers.*

Time and Distance criteria

The main purpose of this counter is to control the reporting frequency according to the vehicle's displacement. So that the unit increases the report frequency when the vehicle is moving (accumulating traveled distance) and decreases it when the vehicle is not moving. This is more efficient than having the unit report by a *time-only* criteria were almost all of the reports triggered when the vehicle is not moving are not relevant.

Note:

The counter can also be configured to follow a *time-only* criteria. To do so, set the *Distance Threshold* parameter to 0 and the unit will only use the *Minimum Report Time* as trigger.

How does this criteria works ?

First, the signal has to be triggered by a distance threshold that tells the activate its associated TD signal whenever the accumulated distance exceeds this value. This *distance-only* scheme has two problems:

1. What if the vehicle goes too fast and/or the distance threshold is low ?. It will set the signal too frequent.
2. What if the vehicle moves too slow and/or the distance threshold is large ?. Or worse, what if it does not move at all?. It will activate the signal very few times or may never activate it.

To solve this problem the *distance-only* trigger is improved with a ***Time and Distance*** trigger that adds *two* controlling parameters for the *Distance threshold*. These parameters are the *Minimum Time* and the *Maximum Time*.

The *Maximum Time* takes control of the signal when the unit has not exceeded the *Distance Threshold* for a long time. So this value ensures a minimum triggering frequency. One way to see this parameter, is that this time will be the signal activation period when the vehicle is stationary.

The *Minimum Time* takes control of the signal when the unit exceeds the *Distance threshold*. Assuring that even when the unit is exceeding the *Distance threshold* it will not activate the signal too frequently, no more frequent than the configured minimum time value. One way of seeing this parameter is that this time will be the report period when the vehicle is moving too fast.

A *Time and Distance* counter is created with the following parameters:

1. Distance Threshold.
2. Minimum Time.
3. Maximum Time.

6.25. (TD) TIME AND DISTANCE SIGNALS CONFIGURATION

The TD message allows to configure the parameters that control the *Time & Distance* signals' trigger. *Ten TD counters may be defined independently.* The message has the following format:

| ABBBB[CCCCDDDEEEE] | | | | |
|--------------------|--------------|--------|-------------|--|
| Chars | Item | Format | Value | Meaning |
| 1 | TD index | A | 0-9 | Time and distance index. |
| 4 | Minimum Time | BBBB | seconds | Minimum amount of time elapsing between reports. To enable just time reporting, this is the only parameter to set, and the others should be set to 0. Setting this value to 0 disables the report. |
| 4 | reserved | CCCC | | This parameter is ignored and can have any value. |
| 4 | Distance | DDDD | x100 meters | Distance the unit must travel between reports. <i>Each unit represents 100 meters.</i> |
| 4 | Maximum Time | EEEE | seconds | Maximum amount of time elapsing between signals' activation. This parameter is only relevant when distance reporting is desired. If set to 0, there is no limit to the amount of time between reports. |

6.26 (TX) Text Message

Qualifiers: S, R

This message is used to transfer messages between the unit and any *Destination*. Printable ASCII characters as well as byte-like messages can be transferred by means of escaping sequences. The message has the following format:

| [A...] | | | | |
|--------|-------------|--------|--------|--|
| Chars | Item | Format | Value | Meaning |
| | Text String | [A] | string | Any printable character but ' ; ' , ' > ' , ' < ' . These characters and any other non-printable characters can be transmitted using the escape sequences of the next table. |

6.26.1 Escape sequences

| Sequence | Meaning |
|----------|--|
| \a | 0x07 (alert) |
| \b | 0x08 (backspace) |
| \e | 0x1B (escape) |
| \f | 0x0C (form feed) |
| \n | 0x0A (line feed) |
| \\ | 0x5C (' \ ') |
| \t | 0x09 (tab) |
| \r | 0x0D (carriage return) |
| \s | 0x3B (' ; ') |
| \y | 0x3E (' > ') |
| \z | 0x3C (' < ') |
| \XX | Any two-digit character hex value, printable or not. |

6.26.2 Garmin Mode Messages

When the Garmin Mode is active (Please consult the **XAGM** TAIP message for information about Garmin Mode), *Antares SBTM* will send the Status Messages and Text Messages to the server using the following modifications of the **TX** TAIP message:

(TXGMKI) Garmin Mode Driver ID - Unsolicited

This message will be sent by *Antares SBTM* each time the user changes the Driver ID on the Garmin device. This message has the following format:

6.26. (TX) TEXT MESSAGE

| AAAAAAAAAABBBBBBBBCCCC... | | | | |
|---------------------------|----------------|------------|-------------|---|
| Chars | Item | Format | Value | Meaning |
| 10 | Change ID | AAAAAAAAAA | 1-999999999 | This counter increases everytime the Driver ID changes. The starting value is 1 |
| 10 | Time of change | BBBBBBBBBB | Seconds | Time when the change was made. Is the number of seconds since 12:00 am December 31, 1989 UTC. The Garmin device will receive this time as GMT(0), therefore the GMT offset must be calculated when using this TAIP message. |
| 20 max. | Driver ID | CCC... | Varies | Driver's ID. |

For example, *Antares SBTM* will send the following message when the Driver ID is changed on the Garmin device:

```
>RTXGMKI00000000020620232668AuthorizedDriver2;ID=Test<
```

(TXGMKS) Garmin Mode Change Driver Status - Unsolicited

This message will be sent by *Antares SBTM* each time the user changes the Driver Status on the Garmin device. This message has the following format:

| AAAAAAAAAABBBBBBBBCCCCCCCCC | | | | |
|-----------------------------|----------------|------------|-------------|---|
| Chars | Item | Format | Value | Meaning |
| 10 | Change ID | AAAAAAAAAA | 1-999999999 | This counter increases everytime the Driver ID changes. The starting value is 1 |
| 10 | Time of change | BBBBBBBBBB | Seconds | Time when the change was made. Is the number of seconds since 12:00 am December 31, 1989 UTC. The Garmin device will receive this time as GMT(0), therefore the GMT offset must be calculated when using this TAIP message. |
| 10 | Status ID | CCCCCCCCC | 1-16 | Status ID to be set. |

For example, *Antares SBTM* will send the following message when the Driver Status is changed on the Garmin device:

```
>RTXGMKS00000000020620233658000000000;ID=Test<
```

(TXGMSS) Garmin Mode Stop Message Status - Unsolicited

This message will be sent by *Antares SBTM* each time the user performs an action on a Stop Message found in the Garmin's Stop Message list. This message will only be sent by *Antares SBTM* if the Garmin device support Application Protocol A603. If the complete Stop Message list is deleted, *Antares SBTM* will only report that the last message on the list was deleted. This message has the following format:

6.26. (TX) TEXT MESSAGE

| AAABBBCCC | | | | |
|-----------|---------|--------|---------|---|
| Chars | Item | Format | Value | Meaning |
| 3 | Stop ID | AAA | 000-255 | ID used to identify the Stop Message to be consulted. |
| 3 | Status | BBB | 100-104 | 100: Active status. The Stop is active. The value of Index will correspond to the current position of the Stop in the list. 101: Done status. The Stop is marked as Done. The value of Index will correspond to the current position of the Stop in the list. 102: Unread Inactive. The Stop has not been read and it is inactive. The value of Index will correspond to the current position of the Stop in the list. 103: Read Inactive. The Stop has been read but it is inactive. The value of Index will correspond to the current position of the Stop in the list. 104: Deleted. The Stop has been deleted. The Garmin device will return this status for any Stop that is not present in the Stop list. The value of Index will be set as "—" and it should be ignored by the server. |
| 3 | Index | CCC | 0-255 | Position of the Stop in the Garmin's Stop List. |

For example, when the user reads a Stop Message in the Garmin device, Antares will send to the server:

```
>RTXGMSS000103000<
```

And if the user sets this Stop Message as active, Antares will send:

```
>RTXGMSS000100000<
```

But if the user deletes the Stop Message, Antares will send this message instead:

```
>RTXGMSS000104---<
```

(TXGMTC) Garmin Mode Canned Reply - Unsolicited

This message will be sent by *Antares SBTM* each time the user selects a Canned Reply that was generated by a Canned Reply Text Message. This message has the following format:

6.26. (TX) TEXT MESSAGE

| AAAAAAAAAABBBBBCCC | | | | |
|--------------------|-----------------|------------|---------|--|
| Chars | Item | Format | Value | Meaning |
| 10 | Time | AAAAAAAAAA | Seconds | Time when the reply was sent. Is the number of seconds since 12:00 am December 31, 1989 UTC. The Garmin device will receive this time as GMT(0), therefore the GMT offset must be calculated when using this TAIP message. |
| 5 | Message ID | BBBBB | 0-99999 | This is the ID of the Canned Reply Text Message that is being replied to. |
| 3 | Canned Reply ID | CCC | 0-200 | ID used to identify each Canned Reply. |

For example, *Antares SBTM* will send the following message which indicates which Canned Reply was chosen:

```
>RTXGMTC062023468800000001;ID=Test<
```

(TXGMTR) Garmin Mode Received Text Message - Unsolicited

This message will be sent by *Antares SBTM* each time the user sends a text message using the Garmin device. This is an unsolicited message and the server should be configured to receive this message. This message will only be sent by *Antares SBTM* if the Garmin device support Application Protocol A602 or A604. The message has the following format:

| AAAAAAAAAABBBBBBBBBB[CCC...] | | | | |
|------------------------------|------------|------------|---------|---|
| Chars | Item | Format | Value | Meaning |
| 10 | Message ID | AAAAAAAAAA | integer | This is the message ID sent by the Garmin device. This ID is auto-incremented by Garmin with each message sent. |
| 10 | Time | BBBBBBBBBB | seconds | This is the Time and Date when the message was generated. Is the number of seconds since 12:00 am December 31, 1989 UTC. This time will reflect the exact time and date set on the Garmin device. |
| 0-50 | Message | [CCC...] | varies | This is the text message sent by the Garmin device screen. It can contain any set of ASCII characters. The messages can be up to 50 characters long. If the message was longer than 50 characters, the message will be truncated. |

Fore example, when the user sends a text message using the Garmin device:

```
Hello World!
```

Antares will send to the server a message like this:

```
>RTXGMTR000000001140936501600Hello World!<
```

6.26. (TX) TEXT MESSAGE

(TXGMTS) Garmin Mode Sent Message Status - Unsolicited

This message will be sent by *Antares SBTM* each time the user performs an action on a Text Message found in the Garmin's Inbox. This message will only be sent by *Antares SBTM* if the Garmin device support Application Protocol A604. If the complete Inbox is deleted, *Antares SBTM* will only report that the last message on the list was deleted. The message has the following format:

| AAAAAB | | | | |
|--------|------------|--------|---------|--|
| Chars | Item | Format | Value | Meaning |
| 5 | Message ID | AAAAA | integer | The ID of the message that was modified. |
| 1 | Status | B | 0-2 | 0: Message is unread. 1: Message is read. 2: Message not found or deleted. |

For example, when the user opens a message from the Garmin's inbox it will be marked as read, and Antares will send a message to notify this action:

>RTXGMTS000001<

If the message is deleted from the inbox, Antares will send to the server:

>RTXGMTS000002<

6.27. (VR) VERSION NUMBER

6.27 (VR) Version number

Qualifiers: Q, R

This message returns the unit's firmware version. The message has the following format:

| Antares GPS [A]A.AA[B] [C] | | | | |
|----------------------------|----------------|--------|-------|--------------------------|
| Chars | Item | Format | Value | Meaning |
| 3 | Version number | A.AA | | Firmware version number. |
| 1 | Reserved | B | | |
| 1 | Reserved | C | | |

6.28 (XAAC) Analog to Digital converter*Qualifiers:* Q, R

Use this message to consult the actual computed value of the ADC. The message has the following format:

| AAAAAPBBB | | | | |
|-----------|----------------------|--------|--------------|---|
| Chars | Item | Format | Value | Meaning |
| 5 | ADC Computed voltage | AAAAA | millivolts | ADC computed value: Average value over a 10 seconds interval of the ADC input voltage. ADC voltage range is 0 - 32V |
| 1 | <i>Indicator</i> | P | <i>fixed</i> | The 'P' character separates voltage from percentage value. |
| 3 | Percent | BBB | percentage | ADC percentage based on the 0 to 32V range. |

6.29 (XAAU) Challenge Text

Qualifiers: Q, R Use this message to send the Challenge Text to *Antares SBTM* in order to authenticate the connection. The message has the following format:

| ABBB... | | | | |
|---------|----------------|--------|--------|---|
| Chars | Item | Format | Value | Meaning |
| 5 | Method flag | A | flag | M: MD5. |
| 4-80 | Challenge Text | BBB... | varies | When sending this message to <i>Antares SBTM</i> , this is any random text that will be used by the Authentication Mechanism. <i>Antares SBTM</i> will reply with the result of encrypting the password and the challenge text. |

For example, the server sends the following message to *Antares SBTM* to begin the authentication process:

>QXAAUMchallenge_text<

Antares SBTM will then reply with the following message:

>RXAAUMoperation_result<

Where operation_result is obtained from the operation:

MD5(MD5(password);challenge_text)

The server must do this same operation and compare the results and take action based on the comparison result. The “;” character must be included in the operation. Please refer to the *Authentication Mechanism* section of the *Operation* chapter for more information.

6.30. (XABS) BATTERY STATUS

6.30 (XABS) Battery Status

Qualifiers: Q, R

Use this message to consult the actual state of the built-in back up battery³. The message has the following format:

| ABBBBPCCC | | | | |
|-----------|-------------------|--------|--------------|--|
| Chars | Item | Format | Value | Meaning |
| 1 | Main Power State | A | flag | Indicates that the main power source (vehicle's battery) is ON (1) or OFF (0). |
| 4 | Battery voltage | BBBB | millivolts | Built-in battery voltage. |
| 1 | Indicator | P | <i>fixed</i> | The 'P' character separates voltage from percentage value. |
| 3 | Percent of charge | CCC | percentage | Built-in battery charge level. |

³ Built-in back-up battery is an optional add-on

6.31 (XACE) Cell Environment

Qualifiers: Q, R

Use this message to query Cellular Network Cell Environment information. The message has the following format:

| A[;BBB...] | | | | |
|------------|------------|--------|-------|---|
| Chars | Item | Format | Value | Meaning |
| 1 | Query type | A | 1-2 | Type of query: 1: Cell ID information as described outside the table. 2: 6-Neighbor Cells information as described outside the table. |

1. Cell ID:

- If Cell Identity is available:
MCC, MNC, LAC, CI, BSIC, BCCH Freq (absolute), RxLvl, RxLvl Full, RxLvl Sub, RxQual, RxQual Full, RxQual sub, Idle TS.
- If Cell Identity is not available:
MCC, MNC, LAC,, BSIC, BCCH Freq (absolute), RxLvl, RxLvl Full, RxLvl Sub, RxQual, RxQual Full, RxQual sub, Idle TS.

2. Neighbor1 to Neighbor2:

- If Cell Identity is available:
MCC, MNC, LAC, CI, BSIC, BCCH Freq (absolute), RxLvl.
- If Cell Identity is not available:
MCC, MNC, LAC,, BSIC, BCCH Freq (absolute), RxLvl.

Where:

- MCC: Mobile Country Code. 3 digits.
- MNC: Mobile Network Code. 3-2 digits.
- LAC: Local Area Code. 4-digit Hexadecimal value identifying a group of cells in a a network
- CI: Cell ID. 4-digit Hexadecimal value.
- BSIC: Base Station Identity Code.
- BCCH: Broadcast Control Channel.
- RxLvl: Average Receive level.
- RxLvl Full: RxLvl of the cell accessed over all TDMA frames. Values
- RxLvl Sub: RxLvl of the cell accessed over a subset of TDMA frames.
- RxQual: Average received signal quality (BER).
- RxQual Full: RxQual of the cell accessed over all TDMA frames.
- RxQual Sub: RxQual of the cell accessed over a subset of TDMA frames.
- Idle TS: Idle Time Slot: Time Slot for listening the control channel.

All Rx Levels are measured from 0 to 63 where

0 = -110dBm

1 to 62 = -109 to -47 dBm

63 = -48dBm.

6.32 (XACR) Counter Report

Qualifiers: S, R

Use this message to have the unit generate a TX message to a *Destination Point* (DP) or *Destination Address* (DA) containing an specific counter's (see GC message) value. The message takes a counter's index, and a DP or DA index. The message has the following format:

| AABB | | | | |
|-------|---------------|--------|-------|---|
| Chars | Item | Format | Value | Meaning |
| 2 | Counter Index | AA | 00-19 | Counter's index whose value is going to be reported on a TX message. |
| 2 | Destination | BB | | Destination for the TX message: If the first character of the field is the letter A it is understood by the unit that the second digit holds a <i>Destination Address</i> . If both characters of the field are digits, the unit assumes the user is specifying a <i>Destination Point</i> . Destination Addresses range from 0 to 9. Destination Points range from 00 to 15. |

6.32.1 Reported Message

After receiving the XACR command the unit will send a TX message to the selected DP/DA with the selected counters value.

The message for a counter configured to count distance is as follows:

```
>RTXDistancer 2 = 10 x(1000m)<
```

This is going to be displayed on the AVL software as:

```
Distancer 2 = 10 x(1000m)
```

The first word holds the type of counter, then the counters index is shown. After the equal value the counters value followed by the delta factor configured for the counter.

Other types of counter messages are:

```
Timer 5 = 2 x(300s)
```

```
Distancer 2 = 70 x(1000m)
```

```
Counter 3 = 1200 x (10)
```

If the counter's index is followed by (S) it means the counter is actually suspended.

For example:

```
Distancer 2(S) = 1200 x(10m)
```

If no (S) is shown, it means the counter is running.

The counter type *Counter* indicates a user defined counter.

See the GC message for more information.

6.33 (XACT) Communication Test

Qualifiers: S, R

Use this message to generate an event code to a *Destination Address* (DA). This message is used to test the communication with a given destination(s) without requiring for an specific event to occur. It is also a method for testing a DA definition and testing the remote host identification process of an **EV** message. The message has the following format:

| Chars | Item | Format | ABB | Meaning |
|-------|---------------------|--------|-------|---|
| | | | Value | |
| 1 | Destination Address | A | 0-9 | Destination Address to which generate the "fake" event. |
| 2 | event code | BB | 00-49 | Event code to report. |

6.34 (XADM) Diagnostic Message

Qualifiers: Q, R

Use this message to query unit's status. The message has the following formats:

For diagnostic level 0:

| ABBCCDEFFGHHIIJKLMMNN | | | | |
|-----------------------|---------------------|--------|-----------|--|
| Chars | Item | Format | Value | Meaning |
| 1 | level | A | | Diagnostic level. 0: First level |
| 2 | FW family | BB | 05 | Firmware family. Always return 05 |
| 2 | FW version | CC | – | Firmware version. |
| 1 | SIM state | D | 0-9 | SIM card state: 0: No PIN attempts left. 3: PIN error. 6: SIM ready. 7: No PIN set. 9: Starting. |
| 1 | GSM status | E | 0-5 | GSM status: 0: Not registered, not searching. 1: Registered, home network. 2: Not registered, searching. 3: Registration denied. 4: Unknown. 5: Registered, roaming. |
| 2 | RSSI | FF | 00-31 | Received Signal Strength Indicator |
| 1 | GPRS Attach | G | flag | '1' indicates that the unit is GPRS attached. '0' indicates GPRS de-attached state. |
| 1 | GPRS state | H | 0-9 | GPRS state: 0: Initializing. 2: Idle. 4: Dialing. 6: Connected (Ready). 7: No APN. 8: Stopped. 9: No network. E: Error. (Use QXANS) |
| 2 | Number of SVs | II | | Number of usable GPS satellites. |
| 1 | GPS date source | J | | Source of GPS data. See Source table on PV message. |
| 1 | GPS data age | K | | Age of GPS data. See Age table on PV message. |
| 1 | GPS tracking status | L | 0-C (hex) | GPS tracking status code. See Tracking Status table on ST message. |
| 2 | GPIOs Mask | MM | (hex) | Always returns F0. |
| 2 | I/Os state | NN | (hex) | Inputs (less significant nibble) and Outputs state. |

For diagnostic level 1:

6.34. (XADM) DIAGNOSTIC MESSAGE

ABBCCDDEFGGHHIJJKLLMNNOOPPQRRR

| Chars | Item | Format | Value | Meaning |
|-------|------------------------------|--------|-----------|--|
| 1 | level | A | | Diagnostic level. 1: Second level |
| 2 | FW family | BB | 05 | Firmware family. Always return 05 |
| 2 | FW version | CC | – | Firmware version. |
| 2 | FW version Type | DD | – | SV: Stable Version. Ax: Alfa version. x indicates the revision number Bx: Beta version. x indicates the revision number |
| 1 | SIM state | E | 0-9 | SIM card state: 0: No PIN attempts left. 3: PIN error. 6: SIM ready. 7: No PIN set. 9: Starting. |
| 1 | GSM status | F | 0-5 | GSM status: 0: Not registered, not searching. 1: Registered, home network. 2: Not registered, searching. 3: Registration denied. 4: Unknown. 5: Registered, roaming. |
| 2 | RSSI | GG | 00-31 | Received Signal Strength Indicator |
| 1 | GPRS Attach | H | flag | '1' indicates that the unit is GPRS attached. '0' indicates GPRS de-attached state. |
| 1 | GPRS state | I | 0-9 | GPRS state: 0: Initializing. 2: Idle. 4: Dialing. 6: Connected (Ready). 7: No APN. 8: Stopped. 9: No network. E: Error. (Use QXANS) |
| 2 | Number of SVs | JJ | | Number of usable GPS satellites. |
| 1 | GPS date source | K | | Source of GPS data. See Source table on PV message. |
| 1 | GPS data age | L | | Age of GPS data. See Age table on PV message. |
| 1 | GPS tracking status | M | 0-C (hex) | GPS tracking status code. See Tracking Status table on ST message. |
| 2 | GPIOs Mask | NN | (hex) | Always returns F0. |
| 2 | I/Os state | OO | (hex) | Inputs (less significant nibble) and Outputs state. |
| 1-2 | Reset Diagnostic destination | PP | varies | Destination configured for the Reset Diagnostic messages. UU: Reset Diagnostic messages are not active. A0-A9: Destination Address configured for the Reset Diagnostic messages. 00-15: Destination Point configured for the Reset Diagnostic messages. |
| 2 | Current DP | QQ | 00-15 | Destination Point that generated the Diagnostic Message query. |
| 3 | Keep Alive time | RRR | 001-999 | Keep Alive time used by the Current DP. 000 indicates that the Keep Alive is not active. |

6.34. (XADM) DIAGNOSTIC MESSAGE

For example, to query, send:

>QXADM0<

The unit responds:

>RXADM0053090310007320F000<

When using 1 as Diagnostic level:

>QXADM1<

The unit responds:

>RXADM10530B690310008320F0001515000<

6.35 (XADP) Destination Points

Qualifiers: Q, S, R

This message configures both IP-type and Telephone destinations.

6.35.1 IP-type destinations

For IP-type destination, i.e. *Destination Points* 00 to 09 use the following format and table:

| AABCD[DDD...];E[EEE...] | | | | |
|-------------------------|---------------------------------|---------|------------|---|
| Chars | Item | Format | Value | Meaning |
| 2 | Destination Point index | AA | 00-09 | IP-Type destination point. |
| 1 | Console access or Delete action | B | | Access flag/Action: U: Delete the <i>Destination Point</i> . 0: The IP-type host has TAIP console access. 1: The IP-type host has no TAIP console access. <i>Error 8 is returned on every message.</i> <i>Other values work but are reserved for future implementations. The user should not use any value different than 0, 1 and U.</i> |
| 1 | TCP/UDP selection | C | flag | TCP/UDP and UDP-ack, UDP-no-ack selection. 0 or 1: This DP works on TCP. 2: UDP <i>without</i> confirmation. 3: UDP <i>with</i> confirmation. |
| variable | IP-host | D[D...] | 1-50 chars | IP address or name of the IP-host. Use the standard dot-separated numbers/names for hosts. Ex: 192.168.0.1 or av1.server.com. |
| variable | TCP/UDP Port | E[E...] | 0-65535 | TCP or UDP port used by the IP server for listening to the unit's reports. |

6.35.2 Telephone destinations

For Telephone destination, i.e. *Destination Points* 10 to 14 use the following format and table:

6.35. (XADP) DESTINATION POINTS

| AABCD[DDD...] | | | | |
|---------------|-------------------------------|---------|------------|---|
| Chars | Item | Format | Value | Meaning |
| 2 | Destination Point index | AA | 10-14 | Telephone destination point. |
| 1 | Type of host or Delete action | B | | DP type/Action: U: Delete the <i>Destination Point</i> . 0: Report messages are sent as TAIP messages to this destination. 1: User-defined messages are sent instead of TAIP. See the XATM message. <i>Other values are reserved for future use.</i> |
| 1 | Access | C | | Access for this Telephone. 0: Full access. 1: TAIP console via SMS restricted, VOICE call reception allowed. 4: TAIP console via SMS allowed, VOICE call reception restricted. 5: Full restriction. |
| variable | Telephone | D[D...] | 1-50 chars | Telephone number. |

For more information and examples refer to the *Configuration* chapter.

6.36 (XAEF) Extended-EV message Formats

Qualifiers: Q, S, R

This message allows the creation and configuration of up to three sets of information tags to be used by an event having the Message ID qualifier set to *A*, *B* or *C*. This will make such an event generate an EV reporting message with extra information tags as described on the EV message section.

| A [BBB...] | | | | |
|------------|------------------|--------|-----------|---|
| Chars | Item | Format | Value | Meaning |
| 1 | Message ID | A | A, B or C | The extended-EV format being set or consulted |
| varies | Information Tags | varies | | Enter the following tags separated by a ';' character: AC: Vehicle acceleration. AL: Vehicle altitude. AD: Analog To Digital Converter. BL: Battery level. CVxx: Counter xx value. IO: Input/Outputs. NS: GSM/GPRS Network state. SV: GPS Satellites in view. CE: Cell ID Information. CF: Cell ID Information (Full). OE: Instantaneous Engine's RPM. <i>Vehicle Dependant</i> OT: Instantaneous Throttle (0-100%). <i>Vehicle Dependant</i> OF: Fuel Level. <i>Vehicle Dependant</i> OG: Remaining fuel gallons. <i>Vehicle Dependant</i> OR: Instantaneous Fuel Rate. <i>Vehicle Dependant</i> OS: OBD Status: Comm State, Check Sum error, Ignition, MIL, Secondary Tool. OD: Vehicle's Odometer in miles. <i>Vehicle Dependant</i> OI: Trip Odometer in .1 miles. <i>Vehicle Dependant</i> VO: Virtual Odometer value. RE: Region Event. 'U': Delete this format. |

For example, to set event 49 to send an extended-EV message that includes the vehicle's acceleration, the number of GPS satellites in view and the state of distance counter 05 whenever the vehicle's speed goes beyond 55 mph: Define the event. Set it to use extended-EV format *A*

```
>SED49NA0;S00+<
```

Define extended-EV format *A* to include the required tags

```
>SXAEEA;AC;SV;CV05<
```

To delete the extended-EV reporting format send

```
>SXAEEAU<
```

6.36. (XAEF) EXTENDED-EV MESSAGE FORMATS

See the *Reports' messages* section on the *Operation* chapter, the *Event Machine* section on the *Configuration* chapter and the **ED** and **EV** TAIP messages for more information. See also the *Scenarios and examples* chapter.

6.37. (XAFU) FIRMWARE UPGRADE (OVER THE AIR)

6.37 (XAFU) Firmware Upgrade (Over the air)

Qualifiers: S, R

This message starts an over-the-air firmware update process. This message returns TAIP error 69 or 90 when used on a unit with this feature disabled. The message has the following format:

| ABBBBCCDD | | | | |
|-----------|-------------------|--------|-------|---|
| Chars | Item | Format | Value | Meaning |
| 1 | Server | A | 0-1 | Firmware server: Use 0 for DCT's servers, 1 for user-defined server ^a . |
| 4 | Firmware Version | BBBB | | Firmware Version to download. The first two characters must always be 05 which indicate firmware family. The last two indicate the firmware version. For example, to update to FW 5.22, BBBB should be set to 0522. |
| 2 | Firmware type | CC | | Type of version to download. Always set this value to SV. |
| 2 | Diagnostics route | DD | | Indicates where to send the diagnostic reports of the download/installation process. Use a number between 00 and 15 to indicate a <i>Destination Point</i> . Use the letter A plus a number between 0 and 9 to use a <i>Destination Address</i> . |

^a Contact Digital Communications Technologies for more information on this.

For example, to update to firmware version 5.22:

>SXAFU00522SV15<

Notice that diagnostic messages will be sent to the serial port (DP 15).

If diagnostic messages have to be sent to *Destination Address 5*, use:

>SXAFU00522SVA5<

See the *Firmware Upgrade* section on the *Operation* chapter for more information.

6.38. (XAGA) ADC LEVELS

6.38 (XAGA) ADC levels

Qualifiers: Q, S, R This message is used to configure analog-to-digital converter levels associated to "D" signals. The message has the following format:

| AABCCCCC | | | | |
|----------|--------------|--------|-------|---|
| Chars | Item | Format | Value | Meaning |
| 2 | ADC level ID | AA | 00-04 | Identification code assigned to the ADC level. |
| 1 | Active flag | B | flag | V: ADC level defined as a voltage value. ADC range is 0-32V. P: ADC level defined as a percentage value. U: Delete the ADC level. |
| 5 | ADC level | CCCCC | | A millivolts value or percentage value. |

See the *Analog to Digital Converter* section on the *Operation* chapter for examples.

6.39 (XAGB) Back-up Battery levels

Qualifiers: Q, S, R This message is used to configure back-up battery levels associated to "B" signals. The message has the following format:

| AABCCCCC | | | | |
|----------|------------------|--------|-------|---|
| Chars | Item | Format | Value | Meaning |
| 2 | Battery level ID | AA | 00-04 | Identification code assigned to the Battery level. |
| 1 | Active flag | B | flag | V: Battery level defined as a voltage value. P: Battery level defined as a percentage value. U: Delete the battery level. |
| 5 | Battery level | CCCCC | | A millivolts value or percentage value. |

See the *Back-up battery* section on the *Operation* chapter for examples.

6.40 (XAGF) Store & Forward Thresholds

Qualifiers: Q, S, R

The Store & Forward Thresholds allow to create reports based on the number of stored events in the Store & Forward Buffer of a particular Destination Point. The message has the following format:

| AABCCDDDD | | | | |
|-----------|-------------------|--------|---------|---|
| Chars | Item | Format | Value | Meaning |
| 2 | Index | AA | 00-04 | Store & Forward threshold index. |
| 1 | Action | B | flag | N: Define the Store & Forward threshold as a total number of stored messages. P: Define the Store & Forward threshold as a percentage value. U: Delete the Store & Forward threshold. |
| 2 | Destination Point | CC | 00-14 | Indicates the Destination Point that will be monitored for messages stored in the Stored & Forward buffer. |
| 4 | Value | DDDD | integer | Value of the Store & Forward threshold. |

6.41 (XAGH) Heading deltas

Qualifiers: Q, S, R This message is used to configure heading changes to be monitored with *J* signals. The message has the following format:

| AABCCC | | | | |
|--------|-------------|--------|---------|--|
| Chars | Item | Format | Value | Meaning |
| 2 | Index | AA | 00-04 | Heading delta index. |
| 1 | Active flag | B | flag | 1: Define the heading delta. U: Delete the heading delta. |
| 3 | Delta | CCC | 005-090 | Heading delta change to be monitored. |

See the *Configuration chapter* for more information on using heading deltas to create a turn-by-turn report.

6.42 (XAGM) Garmin Mode

Qualifiers: Q, S, R Use Garmin devices as communication tools between the driver and the AVL server. The Garmin device allows an *Antares SBTM* user to send and receive text messages from or to a server. It also allows to receive Stop Messages from the server, that can help create routes for the vehicle to follow. *Antares SBTM* can work with any Garmin device that have the Fleet Management Protocol implemented. The Garmin Mode must be enabled for *Antares SBTM* to communicate with the Garmin device. The **XAGM** TAIP message is used to control the Garmin Mode and the communication between *Antares SBTM* and the Garmin device. The message has the following format:

To Enable/Disable the Garmin Mode:

| AB | | | | |
|-------|-------------------|--------|-------|---|
| Chars | Item | Format | Value | Meaning |
| 1 | Action | A | flag | E: Enable Garmin Mode. D: Disable Garmin Mode. |
| 1 | Destination Point | B | 0-F | Unrequested Status Messages and Text Messages from the Garmin Device will be sent to this Destination Point. One character corresponding to the hex value of the Destination Point (i.e. 00-15 = 0-F). When using the Q qualifier to query the state of the Garmin mode, this field will show the Destination Point set, or U if the Garmin mode is disabled. |

The option to disable the Garmin Mode using this TAIP message is only available over the air. To disable the Garmin Mode locally, send the string EXIT.COMMDATA as a single packet through the serial port.

For example, To enable the Garmin mode send to Antares:

>SXAGME0<

Antares will respond with:

>RXAGME0<

Once the Garmin Mode is active, *Antares SBTM* will send any message typed on the Garmin device to the server, using the **TX** TAIP message. For more information on the format used by *Antares SBTM* to send the messages to the server, consult the **TX** message.

The **XAGM** TAIP message (Garmin Mode) uses several modifiers for each function implemented. Each modifier will be explained below.

6.42. (XAGM) GARMIN MODE

Note:

When Garmin sends a message to the server which uses the time field, the value will reflect the same time set on the Garmin device. However, when sending messages from the server, the Garmin device will handle any time value received as GMT 0. Because of this the time value used in the messages sent to Garmin must be calculated using the local GMT offset.

6.42.1 (XAGMI) Consult Garmin Device General Information

Qualifiers: Q, R To consult the Garmin device general information use the I modifier. If this message is used while the Garmin mode is disabled, the message will return all the parameters as 0. This modified TAIP message uses the following format:

| AAAAAAAAAABBBBBCCCCCDEF | | | | |
|-------------------------|-------------------------|------------|--------|---|
| Chars | Item | Format | Value | Meaning |
| 10 | Garmin ESN | AAAAAAAAAA | Varies | This is the Garmin device's ESN (Electronic Serial Number) which is unique for each device. |
| 5 | Product ID | BBBBB | Varies | This is an unique number given to each type of Garmin device (Model type) |
| 5 | Garmin Software version | CCCCC | Varies | Is the software version number multiplied by 100 (e.g. version 3.11 will be indicated by 00311). |
| 1 | A602 | D | flag | Indicates if the Application Protocol A602 is supported. 0: Not supported 1: Supported |
| 1 | A603 | E | flag | Indicates if the Application Protocol A603 is supported. 0: Not supported 1: Supported |
| 1 | A604 | F | flag | Indicates if the Application Protocol A604 is supported. 0: Not supported 1: Supported |

For example, to consult the information of the Garmin device send:

>QXAGMI<

Antares will respond with:

>RXAGMI3547600041008270046111<

6.42.2 (XAGMKI) Garmin Mode Driver ID

Qualifiers: Q, S, R To set or query the current Driver ID on the Garmin device use the KI modifier. This message only works for Garmin devices that support Application Protocol A604. This message uses the following format:

6.42. (XAGM) GARMIN MODE

| AAAAAAAAAABBBBBBBBCCCC... | | | | |
|---------------------------|----------------|------------|--------------|---|
| Chars | Item | Format | Value | Meaning |
| 10 | Change ID | AAAAAAAAAA | 1-4294967295 | This counter increases everytime the Driver ID changes. The starting value is 1. |
| 10 | Time of change | BBBBBBBBBB | Seconds | Time when the change was made. Is the number of seconds since 12:00 am December 31, 1989 UTC. The Garmin device will receive this time as GMT(0), therefore the GMT offset must be calculated when using this TAIP message. |
| 20 max. | Driver ID | CCC... | Varies | Driver's ID. |

For example, to set a new Driver ID, use the following message:

```
>SXAGMKI00000000010620229003AuthorizedDriver<
```

Antares SBTM will respond like this to confirm that the Driver ID was set:

```
>RXAGMKI00000000010620229003AuthorizedDriver;ID=Test<
```

To query the current Driver ID set on the Garmin device, use this message:

```
>QXAGMKI<
```

Antares SBTM will respond like this:

```
>RXAGMKI00000000010620229003AuthorizedDriver;ID=Test<
```

6.42.3 (XAGMKA) Garmin Mode Add Driver Status

Qualifiers: Q, S, R

To add a new Driver Status on the Garmin device use the KSA modifier. This message only works for Garmin devices that support Application Protocol A604. It is possible to store up to 16 Driver Status. This message uses the following format:

| AAAAAAAAAABBB... | | | | |
|------------------|------------------|------------|--------------|--|
| Chars | Item | Format | Value | Meaning |
| 10 | Driver Status ID | AAAAAAAAAA | 0-4294967295 | This is the ID used to identify each Driver Status on the list. This message only works for Garmin devices that support Application Protocol A604. The list only accepts up to 16 Drives Status, but the ID can be any number within the range. The list is sorted in ascending order. |
| 1-20 | Driver Status | BBB... | Varies | Driver Status that will be displayed on the Garmin device. |

For example, to add a new Driver Status on the Garmin device's list, use this message:

```
>SXAGMKA0000000000Resting, on Lunch<
```

Antares SBTM will reponse like this to confirm that the Driver Status was added:

6.42. (XAGM) GARMIN MODE

>RXAGMKS0000000000Resting, on Lunch;ID=Test<

6.42.4 (XAGMKSD) Garmin Mode Delete Driver Status

Qualifiers: S, R To delete a Driver Status on the Garmin device use the KSD modifier. This message only works for Garmin devices that support Application Protocol A604. This message uses the following format:

| AAAAAAAAAA | | | | |
|------------|------------------|------------|-------|---|
| Chars | Item | Format | Value | Meaning |
| 10 | Driver Status ID | AAAAAAAAAA | 1-16 | This is the ID used to identify each Driver Status on the list. |

For example, to delete a Driver Status from the Garmin device's list, use this message:

>SXAGMKSD00000000001<

Antares SBTM will reponse like this to confirm that the Driver Status was deleted:

>RXAGMKSD00000000001;ID=Test<

It is possible to delete all the Driver Status with a single message, please refer to the TAIP message **XAGMX** for more information on this.

6.42.5 (XAGMKS) Garmin Mode Change Driver Status

Qualifiers: Q, S, R To change the Driver Status for the current Driver ID on the Garmin device use the KS modifier. This message only works for Garmin devices that support Application Protocol A604. Use the Q qualifier to query the current Driver Status. This message uses the following format:

| AAAAAAAAAABBBBBBBBCCCCCCCC | | | | |
|----------------------------|----------------|------------|--------------|---|
| Chars | Item | Format | Value | Meaning |
| 10 | Change ID | AAAAAAAAAA | 1-4294967295 | This counter increases everytime the Driver ID changes. The starting value is 1 |
| 10 | Time of change | BBBBBBBBBB | Seconds | Time when the change was made. Is the number of seconds since 12:00 am December 31, 1989 UTC. The Garmin device will receive this time as GMT(0), therefore the GMT offset must be calculated when using this TAIP message. |
| 10 | Status ID | CCCCCCCC | 1-16 | Status ID to be set. |

For example, to change the current Driver Status, use the following message:

>SXAGMKS000000000106202334750000000001<

Antares SBTM will reponse like this to confirm that the Driver Status was changed:

>RXAGMKS000000000106202334750000000001;ID=Test<

To query the current Driver Status:

>QXAGMKS<

6.42. (XAGM) GARMIN MODE

Antares SBTM will reponse like this:

```
>RXAGMKS000000000106202334750000000001;ID=Test<
```

6.42.6 (XAGMR) Garmin Mode Add or Delete Canned Replies

Qualifiers: Q, S, R To add or delete a Canned Reply to the Garmin device use the **R** modifier. This list can not be consulted on the Garmin device. The server must know the IDs that have been set in order to use them. *Antares SBTM* will repond with Error 94 if an Id that has not been set is used. This message only works for Garmin devices that support Application Protocol A604. The message has the following format:

| ABBCCCC... | | | | |
|------------|-----------------|--------|--------|---|
| Chars | Item | Format | Value | Meaning |
| 1 | Action | A | Flag | A: Add a new Canned Reply. D: Delete a Canned Reply. |
| 3 | Canned Reply ID | BBB | 0-200 | ID used to identify each Canned Reply. |
| 1-40 | Canned Reply | CCC... | Varies | Canned Reply that will be displayed on the Garmin device. |

For example, to add a Canned Reply to the Garmin device, use the following message:

```
>SXAGMRA000Package Delivered<
```

Antares SBTM will reponse like this to confirm that the Canned Reply was added:

```
>RXAGMRA000Package Delivered;ID=Test<
```

To delete a Canned Reply from the Garmin device, use the following message:

```
>SXAGMRD003<
```

Antares SBTM will reponse like this to confirm that the Canned Reply was deleted:

```
>RXAGMRD003;ID=Test<
```

6.42.7 (XAGMRS) Garmin Mode Canned Reply Text Message

Qualifiers: S, R To send a text message to the Garmin device to which the driver is only able to reply to using a Canned Reply, use the **RS** modifier. This message only works for Garmin devices that support Application Protocol A604. The message has the following format:

6.42. (XAGM) GARMIN MODE

| AAAAABBBBBBBBBBCDDD...DDD;EEE... | | | | |
|----------------------------------|------------------|------------|---------|---|
| Chars | Item | Format | Value | Meaning |
| 5 | Message ID | AAAAA | 0-99999 | This counter increases everytime the Driver ID changes. The starting value is 1. |
| 10 | Time of change | BBBBBBBBBB | Seconds | Time when the change was made. Is the number of seconds since 12:00 am December 31, 1989 UTC. The Garmin device will receive this time as GMT(0), therefore the GMT offset must be calculated when using this TAIP message. |
| 1 | Message Type | C | Flag | Indicates how the message is handled. 0: Will put the message in the Garmin inbox. 1: Will show the message on screen immediately. |
| 3-18 | Canned Reply IDs | DDD...DDD | 3-18 | IDs of the Canned Replies that will be available. Each ID is 3 characters long. Up to 6 Canned Replies can be used. |
| 1-55 | Text Message | EEE... | 1-55 | Text message to be sent to the Garmin device. |

For example, to send a text message using the Canned Replies with ID 000, 001 and 002, use the following message:

```
>SXAGMRS0000006202344511000001002;Status of package #KJP123456 ?<
```

Antares SBTM will reponse like this to confirm that the text message was received:

```
>RXAGMRS0000006202344511000001002;Status of package #KJP123456 ?;ID=Test<
```

Then, *Antares SBTM* will send the following message which indicates which Canned Reply was choosen:

```
>RTXGMT062023468800000001;ID=Test<
```

6.42.8 (XAGMS) Garmin Mode Stop Message

Qualifiers: Q, S, R

To send a Stop Message from the server to the Garmin device and to consult the status of a Stop Message use the **S** modifier. The Stop Messages are used to inform the Garmin device of a new destination. When the Garmin device receives a Stop from the server, it displays a floating icon indicating that a Stop Message has arrived. By touching this icon the Stops List will be shown and the option to start navigating to the new destination will be available. The coordinates that represent the Stop destination are given in semicircles. This message will only work for Garmin devices that support the Application Protocol A603.

The following formulas show how to convert between degrees and semicircles:

$$\begin{aligned}\text{degrees} &= \text{semicircles} * (180/2^{31}) \\ \text{semicircles} &= \text{degrees} * (2^{31}/180)\end{aligned}$$

If the result of this operation is a number with less than 10 digits, it must be preceded with as many "0" as needed to complete the 10 digits. For example:

Latitude: $25.7827 * 11930464.7 = 307599592$

This number should be typed in for *Antares SBTM* like this: +0307599592

6.42. (XAGM) GARMIN MODE

Longitude: $-0.1197 * 11930464.7 = -1428076$

This number should be typed in for *Antares SBTM* like this: -0001428076

When using the S qualifier to send a Stop Message, use the following format:

AAABBBBBBBBBCCCCCCCCCCCCDDDDDDDDDE[EEE...]

| Chars | Item | Format | Value | Meaning |
|-------|--------------|-------------|--------------|---|
| 3 | Stop ID | AAA | 000-255 | ID used to identify the Stop Message sent to the Garmin device. |
| 10 | Time | BBBBBBBBBB | 0-4294967295 | This will be the Time and Date shown in the message. Is the number of seconds since 12:00 am December 31, 1989 UTC. A value of 4294967295 (HEX 0xFFFFFFFF) represents an invalid time, and the Garmin device will ignore the time and date. The Garmin device will receive this time as GMT(0), therefore the GMT offset must be calculated when using this TAIP message. |
| 11 | Latitude | CCCCCCCCCCC | 0-1073741824 | Indicates latitude in semicircles. North latitudes are indicated with positive numbers. South latitudes are indicated with negative numbers. The first character indicates sign. |
| 11 | Longitude | DDDDDDDDDDD | 0-2147483648 | Indicates longitude in semicircles. East longitudes are indicated with positive numbers. West longitudes are indicated with negative numbers. The first character indicates sign. |
| 1-40 | Text Message | E[EEE...] | Varies | This text indicates the name given to the Stop destination. It can contain any set of ASCII characters. The messages can be from 1 to 40 characters long. |

When using the Q qualifier to consult the status of a Stop Message, use the following format:

6.42. (XAGM) GARMIN MODE

| AAABBBCCC | | | | |
|-----------|---------|--------|---------|--|
| Chars | Item | Format | Value | Meaning |
| 3 | Stop ID | AAA | 000-255 | ID used to identify the Stop Message to be consulted. |
| 3 | Status | BBB | 100-104 | <div>100: Active status. The Stop is active. The value of Index will correspond to the current position of the Stop in the list.</div> <div>101: Done status. The Stop is marked as Done. The value of Index will correspond to the current position of the Stop in the list.</div> <div>102: Unread Inactive. The Stop has not been read and it is inactive. The value of Index will correspond to the current position of the Stop in the list.</div> <div>103: Read Inactive. The Stop has been read but it is inactive. The value of Index will correspond to the current position of the Stop in the list.</div> <div>104: Deleted. The Stop has been deleted. The Garmin device will return this status for any Stop that is not present in the Stop list. The value of Index will be set as “—” and it should be ignored by the server.</div> |
| 3 | Index | CCC | 0-255 | Position of the Stop in the Garmin's Stop List. |

E.G. To send a Stop Message to the Garmin device, send:

```
>SXAGMS0001244753308+0307600189-0957793861DCT<
```

Antares will respond to confirm that the Stop Message was received like this:

```
>RXAGMS0001244753308+0307600189-0957793861DCT<
```

To consult the status of the Stop Message with ID 000, send:

```
>QXAGMS000<
```

Antares will respond like this:

```
>RXAGMSS000102000<
```

Is the Stop Message has not been read and is inactive. Or it could respond:

```
>RXAGMSS000103000<
```

If the message has been read but is inactive.

6.42.9 (XAGMCS) Garmin Mode Change Stop Message Status

Qualifiers: S, R

To change the status of a Stop Message present in the Garmin device's Stop Message List, use the **CS** modifier. This message only works for Garmin devices that support Application Protocol A604. This message has the following format:

6.42. (XAGM) GARMIN MODE

| AAAB | | | | |
|-------|---------|--------|-------|--|
| Chars | Item | Format | Value | Meaning |
| 3 | Stop ID | AAA | 0-255 | ID used to identify the Stop Message to be modified. |
| 1 | Status | B | 1-3 | The status in which the Stop Message will be set. 1: Mark Stop as Done. 2: Mark Stop as Active. This state indicates that the Garmin device will begin tracing the route to the Stop destination. 3: Delete the Stop. |

For example, to set a Stop as active, send:

```
>SXAGMCS0002<
```

Antares will respond to confirm that the command was received like this:

```
>RXAGMCS0002<
```

Antares will also send an unsolicited message indicating that the status of the Stop was changed:

```
>RTXGMSS000100000<
```

6.42.10 (XAGMTS) Garmin Mode Text Send

Qualifiers:

| |
|------|
| S, R |
|------|

To send Text Messages from the server to the Garmin device use the TS modifier. This message uses the following format:

6.42. (XAGM) GARMIN MODE

| AAAAABCCCCCCCC [DDD...] | | | | |
|-------------------------|--------------|-----------|-------------|---|
| Chars | Item | Format | Value | Meaning |
| 5 | Message ID | AAAAA | 00000-99999 | This is a 5 digit integer that identifies each message sent to the Garmin device. Each message must have a different ID. If a message is sent using an ID that is already in use by Garmin, <i>Antares SBTM</i> will return Error 67. |
| 1 | Message Type | B | flag | Indicates how Garmin should handle the message. This flag is only matters on Garmin devices that support the Application Protocol A604. If the Garmin device does not support A604, all the messages will be sent to the inbox. 0: Sends the message to the Garmin device inbox. The user will consult it from there. 1: Shows the message on screen immediately. |
| 10 | Time | CCCCCCCCC | Seconds | This will be the Time and Date shown in the message. Is the number of seconds since 12:00 am December 31, 1989 UTC. The Garmin device will receive this time as GMT(0), therefore the GMT offset must be calculated when using this TAIP message. |
| 0-70 | Text Message | [DDD...] | Varies | This will be the message shown in the Garmin device screen. It can contain any set of ASCII characters. The messages can be up to 70 characters long. |

For example, to send a text message to the Garmin device using ID 00000 send:

```
>SXAGMTS0000001244751579Hello World<
```

Antares will respond to the server:

```
>RXAGMTS0000001244751579Hello World<
```

6.42.11 (XAGMT) Garmin Mode Message Status

Qualifiers: Q, R

To consult the status of a message sent to the Garmin device use the T modifier. This message only works for Garmin devices that support Application Protocol A604. This message has the following format:

6.42. (XAGM) GARMIN MODE

| AAAAAB | | | | |
|--------|----------------|--------|-------------|---|
| Chars | Item | Format | Value | Meaning |
| 5 | Message ID | AAAAA | 00000-99999 | The ID of the message to be consulted. |
| 1 | Message Status | B | 0-2 | Indicates the status of a message in the Garmin device. 0: Message is unread. 1: Message is read. 2: Message not found or deleted. |

For example, to consult the status of the message with ID 00000, send:

>QXAGMT00000<

Antares will respond like this:

>RXAGMTS000000<

If the message is not read. It will respond like this:

>RXAGMTS000001<

If the message is read.

6.42.12 (XAGMTA) Garmin Mode Set Canned Message

Qualifiers: S, R To setup canned messages on Garmin device use the **TA** modifier. The canned messages are used to send quick replies from the Garmin device. Up to 120 canned messages can be set. This message only works for Garmin devices that support Application Protocol A604. This message has the following format:

| AAAB[BBB...] | | | | |
|--------------|--------------|-----------|--------|--|
| Chars | Item | Format | Value | Meaning |
| 3 | Message ID | AAA | 0-120 | ID of the canned message to be set. If a canned message is set with an ID already in use, it will overwrite the stored canned message. |
| 1-50 | Text Message | B[BBB...] | Varies | This is the canned message that will be saved in the Garmin device. It can contain any set of ASCII characters. |

For example, to set a canned message, send:

>SXAGMTA000Acknowledged<

Antares will respond to confirm that the canned message was stored like this:

>RXAGMTA000Acknowledged<

6.42.13 (XAGMTD) Garmin Mode Delete Canned Message

Qualifiers: S, R To delete canned messages from the Garmin device use the **TD** modifier. This message only works for Garmin devices that support Application Protocol A604. This message has the following format:

6.42. (XAGM) GARMIN MODE

| AAA | | | | |
|-------|------------|--------|-------|---|
| Chars | Item | Format | Value | Meaning |
| 3 | Message ID | AAA | 0-120 | ID of the canned message to be deleted. |

For example, to delete the canned message with ID 000, send:

>SXAGMTD000<

Antares will respond to confirm that the canned message was deleted like this:

>RXAGMTD000<

6.42.14 (XAGMX) Delete Fleet Management Protocol Related Data

Qualifiers: S, R To delete any data related to the Fleet Management Protocol on the Garmin device use the **X** modifier. This message uses the following format:

| A | | | | |
|-------|--------|--------|-------|--|
| Chars | Item | Format | Value | Meaning |
| 1 | Action | A | Flag | <ul style="list-style-type: none">0: Delete all stops.1: Delete all text messages.2: Delete the active navigation route.3: Delete all canned messages.4: Delete all canned replies.5: Not in use.6: Delete all Driver ID and Driver Status information.7: Delete all data related to the Fleet Management Protocol. This flag will also disable the Fleet Management Interface on the Garmin device and will disable the Garmin mode on <i>Antares SBTM</i>. |

To delete all information related to the Fleet Management Protocol on the Garmin device, disable its graphic interface and disable the Garmin Mode in Antares, use the following message:

>SXAGMX7<

Antares SBTM will respond with the following message:

>RXAGMX7;ID=Test<

The Garmin device will then restart and the Fleet Management icon will no be shown. To enable the Fleet Management features on the Garmin device again, send the Garmin Mode enable message to *Antares SBTM*. For example:

>SXAGME0<

6.43 (XAGN) Acceleration Limits

Qualifiers: Q, S, R This message is used to configure positive and negative acceleration limits (thresholds). Acceleration thresholds are monitored with the **N** signals.

Warning:

Note that for positive acceleration thresholds, Acceleration Signals are TRUE when the vehicle's acceleration is larger than the threshold. For negative acceleration thresholds, Acceleration Signals are TRUE when the vehicle's acceleration is less than the threshold.

See the **XAGN** message for detailed information. The message has the following format:

| AABCCC | | | | |
|--------|-------------|--------|-----------|--|
| Chars | Item | Format | Value | Meaning |
| 2 | limit ID | AA | 00-04 | Identification code assigned to the threshold. |
| 1 | Active flag | B | flag | 1: Limit is active. U: Delete limit. |
| 3 | Speed limit | CCC | -99 to 99 | Acceleration limit in miles per hour per second. |

See the *Using Acceleration Signals* section on the *Configuration* chapter for examples.

6.44 (XAGP) GPRS Pause

Qualifiers: Q, S, R

Use this message to make the unit close any open TCP sockets, and end the current GPRS session. After 20 seconds the unit will start the GPRS session again. The message has the following format:

| A | | | | |
|-------|-------------|--------|-------|--|
| Chars | Item | Format | Value | Meaning |
| 1 | GPRS Paused | A | 0-1 | Use '1' to start a GPRS pause procedure. |

For example, to initiate a GPRS pause send to the unit:
>SXAGP1<

Depending on the number of open TCP sockets the response is delayed a few seconds.

6.45 (XAGR) Circular Regions

Qualifiers: Q, S, R

Use this message to create up to 100 circular regions based on a center's coordinates and a radius. Use the **XAIR** to create a circular region centered on the actual unit's GPS position. The message has the following format:

| AABCCDDDDDEEEFFFFFFGGGGGGHHHHH | | | | |
|--------------------------------|--------------------|-----------|----------|---|
| Chars | Item | Format | Value | Meaning |
| 2 | index | AA | 00-99 | Region index. |
| 1 | action | B | 1-U | Use 1 to define a region, U to undefine it. |
| 7 | Center's latitude | CCC.DDDD | degrees | Center's latitude. It does include sign. |
| 8 | Center's longitude | EEEE.FFFF | degrees | Center's longitude. It does include sign. |
| 6 | Region radius | GGGGGG | meters | Region radius (min. 50 meters). |
| 6 | reserved | HHHHHH | reserved | Reserved, set to 000000. |

For example, to create a circular region centered at N27,68694 E86,72917 with a radius of 500m, send to the unit:

>SXAGR331+276869+08672910005000000000<

See the *Using Circular Regions* section on the *Configuration* chapter for more examples. See also the **XAIR** command.

6.46 (XAID) IMEI as ID

Qualifiers: Q, S, R

This message tells the unit whether to use or not (defaults to not) its IMEI as ID.

| A | | | | |
|-------|--------|--------|-------|---|
| Chars | Item | Format | Value | Meaning |
| 1 | action | A | 0-1 | Use 1 to set the unit's ID with the IMEI. Use 0 to set the unit's ID to the value set by the user with the ID message. This is the default state |

6.47. (XAIM) IMEI CONSULT

6.47 (XAIM) IMEI consult

Qualifiers: Q, R

This message is used to consult the unit's IMEI (International Mobile Equipment Identity).

| AAAAAAAAAAAAAB | | | | |
|----------------|-----------|--------|-------|-------------------|
| Chars | Item | Format | Value | Meaning |
| 14 | IMEI | A | | IMEI. |
| 1 | IMEI's CS | B | | IMEI's Check Sum. |

6.48 (XAIO) Input, Outputs consult

Qualifiers: Q, R

This message is used to consult the actual state of inputs and outputs.

;XABCD;IEFHJ;GKLM

| Chars | Item | Format | Value | Meaning |
|-------|----------|--------|-------|---|
| 1 | Out 4 | A | flag | State of Output 4 (1=active, 0=inactive). |
| 1 | Out 3 | B | flag | State of Output 3. |
| 1 | Out 2 | C | flag | State of Output 2. |
| 1 | Out 1 | D | flag | State of Output 1. |
| 1 | Input 4 | E | flag | State of Input 4. |
| 1 | Input 3 | F | flag | State of Input 3. |
| 1 | Input 2 | H | flag | State of Input 2. |
| 1 | Input 1 | J | flag | State of Input 1. |
| 1 | Ignition | K | flag | State of vehicle ignition. |
| 1 | Power | L | flag | State of main power source. |
| 1 | 12/24v | M | flag | State of main power source 12/24voltage detector. |

6.49 (XAIP) IP address

Qualifiers: Q, R

This message is used to consult the actual IP address assigned to the unit by the carrier. IP address 0.0.0.0 is returned when the GPRS session is not up. The message has the following format:

| AAABBBCCDDD | | | | |
|-------------|------------|--------|---------|---------------------------------|
| Chars | Item | Format | Value | Meaning |
| 3 | 1st number | AAA | 000-255 | First number of the IP number. |
| 3 | 2nt number | BBB | 000-255 | Second number of the IP number. |
| 3 | 3rd number | CCC | 000-255 | Third number of the IP number. |
| 3 | 4th number | DDD | 000-255 | Fourth number of the IP number. |

6.50 (XAIR) Create Circular Region “here”

Qualifiers: S, R

When this command is given to the unit, a circular region centered at the actual position is created with the radius given. See also the **XAGR** to create Circular Regions manually. The message has the following format:

| AABBBBBB | | | | |
|----------|--------------|--------|--------|--|
| Chars | Item | Format | Value | Meaning |
| 2 | region index | AA | 00-99 | Circular Region index to be set with the actual GPS position having the radius given in B. |
| 6 | radius | BBBBBB | meters | Region's radius. |

For example, to create a circular region on index 34, centered at the actual position and having a radius of 1km, send:

>SXAIR34001000<

This region can be monitored with the region signal **K34**.

See the *Using Circular Regions* section of the *Configuration* chapter for more examples.

6.51 (XAiT) Driving Metrics

Qualifiers: S, Q, R

Use this message to consult the actual vehicle's acceleration, maximum registered positive acceleration, maximum registered negative (braking) acceleration and maximum registered speed. This command allows to consult also the GPS situation of the vehicle when each maximum occurred. Driving Metrics are discussed in the *Configuration* chapter. The message has the following formats depending on the value used for the Action/Response item:

| A [BBBBCCDDDD,EEE,FFFFF,GGG,HHH] | | | | |
|----------------------------------|--------------------|--------|-----------|--|
| Chars | Item | Format | Value | Meaning |
| 1 | Action/Response | A | R,C | Action/Response type. R: Summary: Response to driving metrics summary. C: Clear: Clear maximums and their GPS data |
| 3 | Actual acc. | BBB | [-99,+99] | Actual acceleration in Miles/(h*s) |
| 3 | Max. Pacc. | CCC | [-99,+99] | Max. positive acceleration in Miles/(h*s) |
| 3 | Max. Nacc. | DDD | [-99,+99] | Max. negative acceleration in Miles/(h*s) |
| 3 | Max. Speed. | EEE | [0,999] | Max. registered speed in MPH |
| 5 | Engine's RPM. | FFFFF | [0,16384] | Max. registered Engine's RPM. |
| 3 | Throttle Position. | GGG | [0,100] | Max. registered Throttle Position. Percentage |
| 3 | Fuel Rate. | HHH | [0,] | Max. registered Fuel Rate. Gallons per hour. |

| Chars | Item | Format | Value | Meaning |
|-------|-----------------|----------|--------|--|
| 1 | Action/Response | A | 0-5,C | Action/Response type. C: Clear: Clear maximums and their GPS data 0: Max. Positive Acc.: Query/Response to max. positive acc. (OBD/GPS) 1: Max. Negative Acc.: Query/Response to max. positive acc. (OBD/GPS) 2: Max. Speed: Query/Response to max. registered speed. (OBD/GPS) 3: Max. Engine's RPM. 4: Max. Throttle Position. 5: Max. Fuel Rate. |
| 3,5 | Metric | III [II] | varies | Value of the requested metric. |
| 37 | EV message | JJJ... | | GPS information corresponding to the requested metric, in EV form. See the EV message for details. |

See the *Configuration* chapter for examples.

6.52 (XAKA) Keep Alive

Qualifiers: S, Q, R

This message is used to activate/deactivate a keep-alive to a given IP-type *Destination Point* (see the XADP message). A keep-alive option should be avoided as it increases the transmission cost. Use this option if you need to be able to interrogate the unit at any time and the unit is working behind a NAT. The keep-alive transmission will prevent the NAT from breaking the connection between your server and your unit due to inactivity. The keep-alive packet contains the unit's ID without any TAIP message so your server has to expect these special-non-TAIP messages. This option can be used both on TCP and UDP IP-type destinations. After the Keep Alive is configured, the unit must be reset using the >SRT< message or else the Keep Alive will not work. The message has the following format:

| ABBB[C] | | | | |
|---------|-------------------|--------|------------------------|--|
| Chars | Item | Format | Value | Meaning |
| 1 | Destination Point | A | 0-9 | The IP-type <i>Destination Point</i> index for which the keep-alive is being configured. |
| 3 | Inactivity time | BBB | 001-999 minutes or 000 | Inactivity time in minutes. Use 000 to deactivate the keep-alive mechanism. <i>All DPs but DP 9 default to 000. DP 9 defaults to 60 minutes.</i> |
| 1 | CR/LF | [C] | flag | 1: Include Carriage Return and Line Feed after each Keep Alive packet. 0: Do not include Carriage Return and Line Feed. |

6.53 (XAKL) GPS Back Log

Qualifiers: Q, R

This message is used to retrieve up to the last 60 messages received from the unit's built-in GPS module. The message has the following format:

| [AAAA] | | | | |
|--------|----------------------------------|--------|-----------|---|
| Chars | Item | Format | Value | Meaning |
| 4 | Number of messages/Message Index | AAAA | 0005-0060 | On a query: Optional number of messages to retrieve. If not set, 60 messages will be sent. On a response: The message index of the actual query. |

For example, to retrieve the last 10 messages stored on the GPS back log, use:
>QXAKL0010<

The unit will respond something like:

```
>RXAKL0000:RPV32129+2236273-0840549303521512<
>RXAKL0001:RPV32130+2236262-0840550103521412<
>RXAKL0002:RTM0855177932507200814109100000<
>RXAKL0003:RPV32132+2236262-0840550103321412<
>RXAKL0004:RAL32133+02582-02512<
>RXAKL0005:RPV32134+2236217-0840553103021712<
>RXAKL0006:RPV32135+2236208-0840553902721912<
>RXAKL0007:RPV32136+2236200-0840554502421712<
>RXAKL0008:RPV32137+2236193-0840554901719412<
>RXAKL0009:RPV32138+2236189-0840555100721512<
>RXAKL0010<
```

Where each GPS message is shown after the log's index. The last message shows no GPS message, indicating the total number of messages retrieved and confirming the user command QXAKL0010.

6.54 (XALL) Local Lock

Qualifiers: S, Q, R

This message is used to lock/unlock most configuration messages (S messages) and query messages (Q messages) on the TAIP console for the serial port. A locking operation consists of using this message and a user-defined 4-character code. An unlocking operation is performed by entering the message in unlocking mode with the same code defined on the locking operation. When locked, any set or query message coming from the serial port is rejected with error code 30 with the following exceptions:

| | |
|---------|---------|
| - SXALL | - QRP |
| - SXARD | - QXANS |
| - QXAIM | - QXARD |
| - QXALL | - QVR |
| - QID | - QXAID |
| - QXADM | |

The message has the following format:

| A [BBBB] | | | | |
|--|-----------|--------|---------------|---|
| Chars | Item | Format | Value | Meaning |
| 1 | Operation | A | | Operation: L: Lock. The console will be locked with the code defined on B. U: Unlock. The console will be unlocked. The unlocking code must be provided on B. |
| 4 | code | BBBB | alpha-numeric | Locking/unlocking code. The locking code can not be 0000 |

For example, to lock configuration messages for the serial port:

```
>SXALLL12XY<
```

Now all configuration messages entered over the serial port will return error 30:

```
>SGS0310020<
```

```
>RER30:SGS0310020<
```

To unlock the TAIP console again, use 'U' and the same code provided on the locking process:

```
>SXALLU12XY<
```

6.55 (XAMD) MD5 Check

Qualifiers: Q, R Use this message to check the MD5 checksum value of any string. This message is used mostly as a diagnostics message that helps confirm the MD5 checksum values calculated by *Antares SBTM* and/or the server performing the authentication. This message has the following format:

| AAA . . . | | | | |
|-----------|------|--------|--------|--|
| Chars | Item | Format | Value | Meaning |
| 4-80 | Text | AAA... | varies | Text to be encoded using the MD5 mechanism / MD5 result, 32 characters long. |

6.56 (XANB) Network Band mode

Qualifiers: S, Q, R

Antares SBTM is capable of working on 4 different cellular bands in the following way: With this command the user selects a mode which may be of *mono band* or *dual-band* type. On *dual-band* modes, bands are automatically switched by the module according to network discovery information. A mode change will be effective only after a system reset (>SRT< or power-cycle). The allowed modes are presented in the next table. The message has the following format:

| A | | | | |
|-------|------|--------|-------|---|
| Chars | Item | Format | Value | Meaning |
| 1 | mode | A | 0-6 | Band mode: 0: mono-band mode 850 MHz. 1: mono-band mode 900 extended MHz (900E). 2: mono-band mode 1800 MHz. 3: mono-band mode 1900 MHz. 4: dual-band mode 850/1900 MHz. 5: dual-band mode 900E(extended)/1800 MHz. 6: dual-band mode 900E(extended)/1900 MHz. |

6.57 (XANS) Network Status (GPRS)

Qualifiers: Q, R

Use this message to consult the state of the GPRS session and the state of the TCP sockets of every IP-type *Destination Point*. The message has the following format:

ABB[{:CDEEE}x 10]

| Chars | Item | Format | Value | Meaning |
|-------|-------------------|--------|-------|--|
| 1 | GPRS Attach state | A | flag | GPRS Attach state: 0: De-attached. 1: Attached. |
| 2 | GPRS state | BB | 00-99 | GPRS state: 00: Initializing. 02: Stack ready 04: Dialing 06: Connected 07: APN not set by user. 08: Stopped. 09: No network. 10-13: Internal error. 20: Closing. 88-99: GPRS set up failure. |
| 1 | Destination Point | C | 0-9 | IP-type destination point index for which D and EEE give information. |
| 1 | Socket State | D | 0-2 | Socket State of the DP indicated by C: 0: Closed. 1: Opening. 2: Open. |
| 3 | Retries | EEE | | Socket-open-retries for the given DP. |

Example, the following response:

```
>RXANS106;02000;10000;20000;31040;40000;52000;61001;70000;80000;92000<
```

Indicates:

- The GPRS session is UP
- The TCP socket with *Destination Points* 0, 5 and 9 is open.
- The TCP socket with *Destination Points* 1, 2, 4, 7 and 8 is closed.
- The connection with DP 3 and 6 is in opening process.

6.58. (XAOE) ENGINE'S RPM THRESHOLDS.

6.58 (XAOE) Engine's RPM thresholds.

Qualifiers: Q, S, R

This message is used to configure the RPM threshold levels associated to the “OE” signals. The message has the following format:

| ABCCCC | | | | |
|--------|-----------|--------|---------|---|
| Chars | Item | Format | Value | Meaning |
| 1 | Index | A | 0-4 | Threshold index. |
| 1 | Action | B | flag | 1: Sets the threshold. U: Deletes the threshold. |
| 5 | RPM Value | CCCCC | Integer | RPM threshold value. From 00000 to 16384 RPM |

For example, to set RPM threshold 3 to 5000 RPM, use:

```
>SXAOE3105000<
```

Now, to generate an event when such RPM threshold is exceeded:

```
>SED30NV0;OE3+<
```

6.59. (XAOF) FUEL LEVEL PERCENTAGE VALUES.

6.59 (XAOF) Fuel Level percentage values.

Qualifiers: Q, S, R

This message is used to configure the fuel level thresholds associated to the “OF” signals. This fuel level thresholds are presented in a percentage value. The message has the following format:

| ABCCC | | | | |
|-------|------------|--------|---------|---|
| Chars | Item | Format | Value | Meaning |
| 1 | Index | A | 0-4 | Threshold index. |
| 1 | Action | B | flag | 1: Sets the threshold. U: Deletes the threshold. |
| 5 | Fuel level | CCC | 000-100 | Fuel Level threshold. In a percentage value. |

For example, to set Fuel Level percentage threshold 1 to 5 percent, use:

```
>SXAOF11005<
```

Now, to generate an event when Fuel Level goes bellow 5 percent:

```
>SED32NV0;OF1-<
```

6.60. (XAOG) REMAINING FUEL GALLONS THRESHOLDS.

6.60 (XAOG) Remaining Fuel Gallons thresholds.

Qualifiers: Q, S, R

This message is used to configure the fuel level thresholds associated to the “OG” signals. This fuel level thresholds are presented in Gallons remaining. The message has the following format:

| ABCCC | | | | |
|-------|------------|--------|---------|---|
| Chars | Item | Format | Value | Meaning |
| 1 | Index | A | 0-4 | Threshold index. |
| 1 | Action | B | flag | 1: Sets the threshold. U: Deletes the threshold. |
| 5 | Fuel level | CCC | 000-500 | Fuel Level threshold. Gallons remaining. |

For example, to set Fuel Level threshold 1 to 10 gallons remaining, use:

```
>SXAOG11010<
```

Now, to generate an event when Fuel Level goes bellow 10 gallons remaining:

```
>SED33NV0;OG1-<
```

6.61 (XAOR) Fuel Rate thresholds

Qualifiers: Q, S, R

This message is used to configure the fuel rate thresholds associated to the “OR” signals. This fuel rate thresholds are presented in Gallons per hour. The message has the following format:

| ABCCC | | | | |
|-------|------------|--------|-------|---|
| Chars | Item | Format | Value | Meaning |
| 1 | Index | A | 0-4 | Threshold index. |
| 1 | Action | B | flag | 1: Sets the threshold. U: Deletes the threshold. |
| 5 | Fuel level | CC | 00-30 | Fuel Level threshold. Gallons per hour. |

For example, to set Fuel rate threshold 1 to 5 gallons per hour, use:

>SXAOR1105<

Now, to generate an event when the Fuel consumption goes above 5 gallons per hour:

>SED34NV0;OR1-<

6.62 (XAOS) OBD Status Consult

Qualifiers: Q, R

Use this message to consult the Vehicle-Interface-Antares communication state, the vehicle's supported OBD variables and the OBD variables' values. The message has the following format:

| A[B[:CC[:DDD...]];EE[:FFF...]]... | | | | |
|-----------------------------------|----------------------|--------|--------|---|
| Chars | Item | Format | Value | Meaning |
| 1 | Consult Type | A | 0-1 | 0: Ask the Communication state and if the communication is working, get the supported parameters. 1: Same as 0 but, if communication is working, add the OBD variables' current values. |
| 1 | Communication status | B | 0-5 | Status: 0: Serial Port not in OBD mode: The user has not enter >SMT0< 1: Initializing. 2: Can not communicate with the Interface. 3: Detecting vehicle. 4: Busy. 5: Communicating with Interface (OK). Updating OBD values. |
| 2 | Supported Parameter | CC | Hex | Hex code representing the supported parameter: 00: Speed. 01: Engine Speed (RPM). 02: Throttle position. 03: Odometer. 04: Fuel Level. 05: Remaining Gallons. 08: Ignition Status. 09: Malfunction Indicator Light (MIL). 0C: Fuel Rate. 22: Trip Odometer. 99: Secondary Tool Status. |
| Varies | Parameter Value | DDD... | Varies | Value of the supported parameter identified by CC. Varies depending of the parameter. |

For example, when using

>QXA0S0<

Antares may respond:

>RXA0S00<

This indicates that Antares is not in OBD mode therefore it can't communicate with the Interface.

If communication with Interface is working, the same query may result in:

6.62. (XAOS) OBD STATUS CONSULT

>RXAOS05;00;01;02;08;09;22;99<

Indicating that communication is OK and the parameters that can be updated from the vehicle are Speed, Engine Speed, Throttle position, Ignition, MIL, Trip Odometer and Secondary Tool status. All other parameters are not supported by the Interface model connected to *Antares SBTM* when reading a particular vehicle.

Furthermore, using:

>QXAOS1<

will return something like:

>RXAOS15;00:000;01:00823;02:013;08:1;09:1;22:000425;99:0<

Indicating:

Speed: 0mph

Engine Speed: 823 RPM

Throttle at 13%

Vehicle is ON

MIL is ON.

Trip Odometer at 42.5 miles

No secondary tool connected.

6.63 (XAOT) Throttle Position thresholds.

Qualifiers: Q, S, R

This message is used to configure the Throttle Position thresholds associated to the “OT” signals. The message has the following format:

| ABCCC | | | | |
|-------|-------------------|--------|---------|---|
| Chars | Item | Format | Value | Meaning |
| 1 | Index | A | 0-4 | Threshold index. |
| 1 | Action | B | flag | 1: Sets the threshold. U: Deletes the threshold. |
| 5 | Throttle Position | CCC | 000-100 | Throttle Position thresholds. Percentage value. |

For example, to set throttle threshold 2 to 70 percent, use:

>SXAOT21070<

Now, to generate an event when such threshold is exceeded:

>SED31NV0;OT2+<

6.64 (XAPM) Power Management

Qualifiers: S, Q, R

This command activates/deactivates the unit's power saving mode (sleep mode). When power saving mode is activated, the unit will go to sleep after 2 minutes as long as no communication activity is detected over the GPRS network, SMS channels or local TAIP console. If there is communication activity when the 2 minutes expire, the unit will wait for 1 minute of communication inactivity before going to sleep.

When power saving is deactivated the unit will never go to sleep and will cancel any previously defined power saving activation timer.

When the unit reaches sleep mode, a low power consumption state is reached on which only an input or a previously user defined timer expiration can make the unit go back to normal mode. On sleep state, an input-change detection or the wake-up timer count are the only functionality available.

This command includes as optional field an input mask to indicate which inputs to monitor when in sleep mode. If no input mask is specified, all inputs are monitored, meaning any input change may awake the unit. A wake-up-time optional field is included to indicate the unit how long to stay in normal mode after awakening before going to sleep again. If not specified or if the special string DDDD is entered, the default value of 2 minutes is used. The communication time-out used when the unit is ready to go to sleep while it has detected over-the-air communication activity is also an optional field that defaults to 1 minute. The last optional field is the wake-up-interval which is used to awake the unit on a time basis. If this field is not included the unit will not wake up on a time basis. The message has the following format:

6.64. (XAPM) POWER MANAGEMENT

A [[[BBBBBCDEFGHI] JJJJ] KKKK] LLLL]

| Chars | Item | Format | Value | Meaning |
|-------|--------------------|--------|-------|--|
| 1 | Action | A | | Action: A: Activate the power saving mode. E: End any active power saving mode. I: Go to power saving mode immediately(*). |
| 5 | Flags | BBBBB | | Reserved flags. Should be set to 11111 |
| 1 | Power | C | flag | Set to 1 to monitor the power detector input when in sleep mode. Set to 0 to ignore this input on sleep mode. |
| 1 | 24v/12v | D | flag | Ignored. i.e. Always set to '0'. |
| 1 | Ignition | E | flag | Set to 1 to monitor the ignition. |
| 1 | IP4 | F | flag | Set to 1 to monitor input 4. |
| 1 | IP3 | G | flag | Set to 1 to monitor input 3. |
| 1 | IP2 | H | flag | Set to 1 to monitor input 2. |
| 1 | IP1 | I | flag | Set to 1 to monitor input 1. |
| 4 | Wake up time | JJJJ | | This is the Wake-up time that tells the unit how long to stay awake after returning from sleep mode. If the field value is entered as DDDD the default time of 2 minutes is used. Wake up time is an integer value that can accept the scalar factors m or h on the least significant field to indicate minutes or hours, if no scalar factor is included, a value of seconds is assumed. <i>The minimum value is 1 minute, the maximum is 999h (999 hours).</i> |
| 4 | Inactivity timeout | KKKK | | This is the communication inactivity timeout in seconds, used when the unit is ready to go to sleep while it has detected over-the-air communication activity. If the field value is entered as DDDD the default time of 1 minute is used. This is an integer value that does not support scalar factors. <i>The minimum value is 60 seconds; the maximum is 3600 seconds.</i> |
| 4 | Wake up interval | LLLL | | This is the wake-up interval used by the unit to return from sleep to normal mode on a time basis. If this field is not included, the unit will not wake-up on a time basis and only an input change may get the unit back to normal mode. The unit will wake up when the elapsed time since entering sleep mode equals the wake-up interval. This is an integer value that can accept the scalar factors m or h on the least significant field to indicate minutes or hours, if no scalar factor is included, a value of seconds is assumed. <i>The minimum value is 1 minute; the maximum is 255h (255 hours).</i> |

(*) When going to sleep immediately using the 'I' qualifier the saving mode is not preserved. i.e. after the unit wakes up the unit ends the power saving mode.

6.64.1 Examples

- 1 Activate the power saving mode. Only the vehicle's ignition may awake the unit, and when awoken, the unit must remain up for 20 minutes:
>SXAPMA000000010000020m<

- 2 Activate the power saving mode. Any input but the 24v/12v detector may awake the unit, and the unit has to awake every 2 hours. Use default values for the rest.
>SXAPMA000001011111DDDDDDDD002h<

- 3 To configure the unit to go to sleep mode any time the external power is removed and to go back to normal mode when the power is connected again:

1. Create an event triggered when the power signal (F13) goes low that activates the Power Saving mode; making sure the Power input is included on the signals to be monitored while on sleep state:

>SED00SV0;F13-;ACT=SXAPMA<

No input mask specified, so any input, particularly the power detector will drive the unit back to normal mode.

2. Create an event triggered when the power signal goes high that deactivates the power saving mode:

>SED01SV0;F13+;ACT=SXAPME<

Having this configuration the unit will go to sleep mode 2 minutes after disconnecting the primary power source.

6.65 (XAPW) Set Password

Qualifiers: Q, S, R Use this message to configure a password for the Authentication Mechanism. The message has the following format:

| ABBBB . . . | | | | |
|-------------|-------------|----------|--------|---|
| Chars | Item | Format | Value | Meaning |
| 1 | Active flag | A | flag | 0: No password has been set. 1: A password is set. The actual password can not be consulted. |
| 4-40 | Password | BBBBB... | varies | Password to be used by the authentication mechanism. |

For example, to set a password, use the following configuration:

```
>SXAPWpassword<
```

To consult if a password is set, use the following message:

```
>QXAPW<
```

Antares SBTM will respond like this:

```
>RXAPW1<
```

Indicating that a password is set, but it will never show the actual password. Please refer to the *Authentication Mechanism* section of the *Operation* chapter for more information.

6.66 (XARD) Reset diagnostics*Qualifiers:* S, Q, R

This messages activates/deactivates the system's resets' information displayed with TX messages to a selected DA or DP. The message has the following format:

| AA | | | | |
|-------|-------------------|--------|-------|---|
| Chars | Item | Format | Value | Meaning |
| 2 | DA or DP / Action | AA | | Command: U: Deactivate the resets' diagnostics. A0-A9: Route diagnostic TX messages to the DA specified by number. 00-15: Route diagnostic TX messages to the DP specified by number. TD: Route diagnostic TX messages to the current DP. The DP used to send this message to <i>Antares SBTM</i> is considered as the current DP. |

6.67. (XARE) REGIONS REPORT

6.67 (XARE) Regions Report

Qualifiers: Q, R

Use the Region ID report to define a single event for several circular and polygonal regions. A report will be sent using only one event code when the associated regions' signals transition. This TAIP message will generate a report every time the unit enters a region and/or the unit leaves a region. The generated report will include: In/Out indicator, the type of region, and the region index. The message has the following format:

| ABBCDE | | | | |
|--------|---------------------|--------|-------|--|
| Chars | Item | Format | Value | Meaning |
| 1 | Action | A | flag | Indicates how the event will be triggered: I: Will only be triggered when the unit goes inside a defined Region. O: Will only be triggered when the unit goes outside a defined Region. B: Will be triggered both when the unit goes into a Region or leaves the Region. U: Undefine the report. |
| 2 | Event Code | BB | 00-49 | Event code that will be used by the generated report. |
| 1 | Event Handling | C | flag | Message routing: N: Normal. Route the <i>Event Message</i> to the specified <i>Destination Address</i> (DA). X: Serial Port. Route the <i>Event Message</i> to the unit's serial port only. |
| 1 | Message ID | D | flag | Generate event message: V: EV message T: ET message O: Orbcmm's SC-Originated Default Message |
| 1 | Destination Address | E | 0-9 | Destination Address where the report will be sent. |

For example, to define the Region ID report using the event code 00 that will report both when the unit goes In or Out from a region, use the following configuration message:

```
>SXAREB00NV0<
```

A report generated when the unit entered the circular region 10 would be:

```
>REV001524377378+0475230-0740249100000012;RE=IK10;ID=test<
```

A report generated when the unit left the polygonal region 20 would be:

```
>REV001524377378+0475230-0740249100000012;RE=OR20;ID=test<
```

For more information on the Region ID reports, refer to the XARE TAIP message.

6.68 (XASD) Destinations' Set

Qualifiers: Q, R

Use this message to query which *Destination Points* and *Destination Addresses* are set on the unit. This is useful for knowing which DPs and DAs are defined, preventing a one-by-one query with the *XADP* and *DA* messages. The message has the following format:

It returns a string which contains a P plus a two-digit number for each DP set and an A plus a two-digit number for each DA found. It also returns the DP that generated the query message.

For example:

>QXASD<

May return:

>RXASD;P00P01P10P11P14;A00A01A02;P00<

Indicating that DPs 00, 01, 10, 11 and 14 are defined, DAs 0, 1 and 2 are defined and that the >QXASD< message was sent from DP 00

See the *XADP* and *DP* messages for more information.

6.69 (XASF) Store & Forward Buffer

Qualifiers: Q, R

Use this message to consult the reports' buffer state of a given *Destination*. The message has the following format:

| AA[BBBBB;CCCCC;DDDDD] | | | | |
|-----------------------|-------------------|--------|---------|---|
| Chars | Item | Format | Value | Meaning |
| 2 | Destination Point | AA | 00-14 | Destination Point. |
| 5 | Total capacity | BBBBB | integer | Total buffer capacity expressed on events. |
| 5 | Used space | CCCCC | integer | Number of events accumulated on the buffer. |
| 5 | reserved | DDDDD | integer | Reserved for technical support. |

6.70 (XASG) SMS Messages Gateway

Qualifiers: Q, S, R The SMS Messages Gateway allows *Antares SBTM* to send any text message through a TAIP message. *Antares SBTM* will retry to send the message up to three times, if it fails to send it the message will be canceled. The message has the following format que using the S qualifier:

| ABB...;CCC... | | | | |
|---------------|--------------------------------|--------|---------|---|
| Chars | Item | Format | Value | Meaning |
| 1 | Action | A | flag | P: Send SMS to a defined Destination Point. N: Send SMS to a phone number. C: Cancels the SMS message before is sent. |
| 2-20 | Destination Point/Phone Number | BB... | integer | For Destination Points 10 to 15. Phone numbers can be up to 20 characters. |
| 1-80 | Text Message | CCC... | varies | Text message to be sent through the SMS Gateway. |

The message has the following format que using the Q qualifier:

| A | | | | |
|-------|-------|--------|-------|---|
| Chars | Item | Format | Value | Meaning |
| 1 | State | A | 0-4 | 0: Message Sent. 1: Message pending to be sent. Retrying. 2: Message was not sent due to a parameter error. 3: Message not sent. Retry timeout. 4: Message not sent. User canceled. |

For example, to send a SMS message to Destination Point 10, use this message:

```
>SXASGP10;Alert!<
```

Or to send a SMS message to a phone number that is not defined in any Destination Point use the following message:

```
>SXASGN3055551234;Return to base<
```

Only one SMS message can be sent at a time using this mechanism. To cancel a pending SMS message that could not be sent for any reason, use the following message:

```
>SXASGC<
```

6.71. (XASI) IMSI CONSULT

6.71 (XASI) IMSI Consult

Qualifiers:

| |
|------|
| Q, R |
|------|

Use this message to consult the unit's International Mobile Subscriber Identity (IMSI). The message has the following format:

| [AAA...] | | | | |
|----------|------|--------|--------|--|
| Chars | Item | Format | Value | Meaning |
| | IMSI | A | varies | International Mobile Subscriber Identity |

6.72 (XATA) SMS Alias

Qualifiers: S, Q, R

The SMS Alias were created to associate a up to ten character long text message to a regular TAIP message. When an Alias is received through a SMS message, the associated command to this Alias is executed by *Antares SBTM*. The message has the following format.

| AAB;CCC...;DDD... | | | | |
|-------------------|--------------|--------|--------|--|
| Chars | Item | Format | Value | Meaning |
| 2 | Index | AA | 00-09 | Store & SMS Alias index. |
| 1 | Action | B | flag | 1: Define the SMS Alias. U: Undefine the SMS Alias. |
| 80 max. | TAIP Message | CCC... | string | TAIP message to be executed. To include a “;” character in a TAIP message use the “\3B” character. |
| 1-10 | Alias | DDD... | string | Text message associated with the TAIP message to be executed. |

For example, to create a SMS Alias that will set the Output 1 of *Antares SBTM* to true use this message:

```
>SXATA001;SSSXP11;VehicleOFF<
```

For this example, Output 1 is connected to a Engine Turn OFF Circuit that will turn the vehicle off when the Output 1 is set to true, that is why the text message VehicleOFF was used.⁴

⁴ Shutting an engine off without considering a vehicle’s state is not advised.

6.73 (XATD) Current Destination Point

Qualifiers: Q, R

Use this message to know the DP index used on the actual communication channel. The command will return the DP index used to generate the QXATD query. The message has the following format:

| AA | | | | |
|-------|----------|--------|-------|--|
| Chars | Item | Format | Value | Meaning |
| 2 | DP index | AA | 00-15 | AA holds the DP index used for the actual communication. |

6.74 (XATM) User-defined Text Messages

Qualifiers: S, Q, R

Use this message to set the user-custom messages to be used when an SMS report is to be generated to a Telephone-type *Destination* configured to receive non-TAIP messages. See the **XADP** message for more information. One user-text message can be created for each event (00 - 49) so that when an event report is to be sent as a user-custom message the sent message is the one defined with this command (XATM) for the event code that generates the report. The message has the following format:

| AA[BBB...] | | | | |
|------------|-----------------------|--------|--------|--|
| Chars | Item | Format | Value | Meaning |
| 2 | Event Code | AA | 00-49 | Event code for which the user-text message defined on BBB... applies to. |
| varies | Text Message / Delete | BBB... | string | Text Message: Any character but '<', '>' and ';' can be used on the string. <i>The maximum size is 50 characters.</i> If this parameter is not present when using the S qualifier, the user-message for event AA gets deleted. |

See the *Adding SMS reporting* example on the *Scenarios And Examples* chapter for more information.

6.75 (XATS) TAIP Console Sniffer

Qualifiers: S, R

Use this message to activate/deactivate the debugging tool that throws to the serial port all of the TAIP console messages exchanged over any communication media. This is useful to check the communication between an IP-host software or Telephone and the unit.

When active, the unit sends any incoming/outgoing TAIP message to the serial port preceded by a **XATS** response with a direction-of-message indicator (I for Ingoing, O for Outgoing), the direction character is followed by a dash and a number indicating the channel on which the message was sent/received.

The message has the following format:

| A[-BB(B)] [CCC...] | | | | |
|--------------------|-----------------------------|--------|-------|--|
| Chars | Item | Format | Value | Meaning |
| 1 | Action/ Direction indicator | A | flag | When using the S qualifier: Use '1' to activate the sniff, '0' to deactivate For R responses: 'I' indicates that the following displayed message was an incoming message. 'O' indicates an outgoing message. 'A' indicates that the message is a valid UDP ACK. 'a' indicates that the message was received via UDP but was not recognized. |
| 3 or 2 | Channel | BB(B) | | Communication channel: 000 to 014 indicate <i>Destination Points</i> 00 to 14 respectively. 255 Indicates serial port and 254 indicates an internal message caused by an event action. If an ACK was received via the UDP network while in Client Mode, it will indicate the DP from which the ACK was received. If an ACK was received via the UDP network while in Server Mode this field will be only 2 digits long and its value will be "US". When a package that is not an ACK is received while in server mode the value will be 016. |
| 1 to 20 | Message | CCC... | | This field will only be used when the package is received via UDP. It contains a possible UDP ACK. If it is recognized as a valid UDP ACK the "Direction indicator" qualifier's value will be "A". If it is not recognized as a valid UDP ACK then the value of the "Direction indicator" qualifier will be "a". If the package is larger than 20 bytes, it will only show the first 20 bytes. |

6.75.1 Example

Activate the TAIP sniffer:

```
>SXATS1<
```

Now, if the IP-type DP 03 (channel 004) sends a >QGS00< to the unit, the unit will send to the serial port the following messages:

```
>RXATSI-004<>QGS00<>RXATSO-004<>RGS00U<
```

Notice the underlined indicators: They show that the incoming message from channel 004 was >QGS00< and the outgoing message to the same channel was >RGS00U<.

6.76 (XAUN) UDP Network

Qualifiers: S, Q, R

An UDP-Network can be set as a validation mechanism for UDP queries originated to the unit's UDP-Server Port configured with the **XAUP** message. The UDP-Network validation for the UDP-Server port is optional as the validation is also done by looking for a match on the IP-type, UDP *Destinations* defined as IP-numbers⁵.

When an UDP-Network is set, the validation for UDP queries received on the UDP-Server port starts with it. If a match is found, a response is sent to the peer. If no UDP-Network match is found or if an UDP-Network is not defined, a search for a match is done on the IP-type, UDP-set *Destinations*, if a match is found, a response is generated to the *UDP destination*⁶. Finally if no match is found, the query is silently discarded.

The UDP-Network is defined by a network's IP and a mask. The network's IP is any IP address that defines the common IP addresses-values among the members of the UDP-Network. The mask is defined as the number of *ones* from left to right which define the host-address portion to validate on a peer's IP address.

The message has the following format:

| A [AA . AAA . AAA . AAA] / B [B] | | | | |
|----------------------------------|----------------|--------|---------|--|
| Chars | Item | Format | Value | Meaning |
| 3 | 1st number | AAA | 000-255 | First number of the IP number. Use 'U' to undefine the UDP-Network (Default) or enter any IP address that defines the UDP network. |
| 3 | 2nd number | BBB | 000-255 | Second number of the IP number. |
| 3 | 3rd number | CCC | 000-255 | Third number of the IP number. |
| 3 | 4th number | DDD | 000-255 | Fourth number of the IP number. |
| 2 | Network's mask | EE | 0-32 | Number of set-bits from left to right that define the mask. |

For example, to set the UDP network as 172.29.1.x, use:

```
>SXAUN172.29.1.0/24<
```

To set the UDP network as 192.168.x.x, use:

```
>SXAUN192.168.0.0/16<
```

To tell the unit not to use an UDP network validation:

```
>SXAUNU<
```

On this last example, the UDP-Server port validation will still be done by the IP-type, UDP-destinations single-match mechanism.

See also the **XAUP** message.

⁵ UDP-Server port validation can only be done on IP-type destinations defined as IP numbers but not with addresses defines with names.

⁶ See the footnotes on the **XAUP** message

6.77 (XAUO) UDP Origin Port

By default *Antares SBTM* chooses the value of the UDP-origin-port when sending UDP datagrams. Usually after a GPRS session has been established, the chosen port is 1024. This value is incremented and reused according to the unit's TCP/IP stack. There are some situations where it is desired no to have the unit chose this value. With the **XAUO** command, the user can set the origin UDP port to any value. The accepted values range from 1 to 65535 and if the special value of 0 is used (default value) the unit is instructed to freely chose the origin port. Notice that fixing the UDP origin port has nothing to do with the UDP Server Port (**XAUP**): The fixed UDP-origin port will not listen for queries until the unit generates an UDP datagram. Meanwhile the UDP Server Port is always listening. Also notice that the UDP origin port of the datagram containing the response to a query sent to the unit throwout the UDP server port will not have as origin port the value set with the **XAUO** command, but the value of the UDP server port (**XAUP**).

The message has the following format:

| AAAAA | | | | |
|-------|-------------|--------|-------------|---|
| Chars | Item | Format | Value | Meaning |
| 5 | Origin Port | AAAAA | 00000-65535 | UDP Origin Port: 'A' sets the originating port for UDP datagrams sent by the unit. When the special value of 00000 is used, the unit does not use a fixed value but dynamically changes it according to internal mechanisms. The default value is 00000, meaning that by default the unit will use a dynamic origin port. |

6.78 (XAUP) UDP Server Port

Qualifiers: S, Q, R

This command is used to configure a listening UDP port on the unit. If the special value 00000 is used the unit won't listen for UDP messages on a fixed port, instead it will only use the dynamic port created when it sends reports to a UDP destination.

When set, the unit listens for UDP messages on this port originated from any IP-type *Destination* configured to work on UDP. This validation can only be done when the *Destination* is defined as an IP number, i.e. a host name can not be validated when trying to communicate through the unit's UDP server port.

Another type of IP-address validation can be done by configuring on the unit an UDP-Network which is allowed to interact with the UDP-Server port. This is achieved with the XAUN message. When an UDP network is set, the unit uses it first to validate an incoming query on the UDP-Server port, if the peer's IP address matches the UDP-Network, a response is sent too the peer⁷. If it does not match with the UDP-Network, it tries to find a match on any of the IP-type, UDP-defined *Destinations*, if a match is found, a response is generated to the IP/UDP-Port defined on the matching *Destination*⁸. Finally, if no match is found, the query is silently discarded.

Regardless of this parameter being set or not, the unit always listens on the dynamic UDP port generated when a report is transmitted to a UDP destination.

The message has the following format:

| AAAAA | | | | |
|-------|-------------|--------|-------------|---|
| Chars | Item | Format | Value | Meaning |
| 5 | Server Port | AAAAA | 00000-65535 | Server Port: 'A' sets the listening UDP port for the unit. Using 00000 makes the unit not to listen on a fixed listening port, it will continue listening on the dynamic UDP port created when UDP reports are transmitted. |

⁷ The response is sent by interchanging Source Port with Destination Port

⁸ Source and Destination ports are not interchanged: An UDP datagram is generated having Source Port randomly-set and Destination Port set to the value configured with the XADP message corresponding to the matching *Destination*

6.79 (XAVC) Voice Call Start

Qualifiers:

| |
|------|
| S, R |
|------|

This command is used to initiate a voice call with a Telephone-type *Destination Point* (DPs 10 to 14). Two optional fields are provided to specify a retry number and a time between retries for failing calls. The unit will start to dial as soon as this command is entered and the selected *Destination* is found to have any value set (a telephone number). The message has the following format:

| AA [BCC] | | | | |
|----------|-------------------|--------|--------------|---|
| Chars | Item | Format | Value | Meaning |
| 2 | Destination Point | AA | 10-14 | Index of the <i>Destination Point</i> which holds the number to dial. |
| 1 | Retry attempts | B | 1-9 | Number of call attempts before giving up dialing. Defaults to 1. |
| 2 | Retry delay | CC | 5-99 seconds | Time between successive retries. Defaults to 5 seconds. |

See the *Adding Voice Interaction* example on the *Scenarios And Examples* chapter for more information.

6.80 (XAVE) Voice Call End

Qualifiers: S, R

Use this message to hang-up any current voice call. If no voice call is being held, the command returns error 65 (>RER65: SXAVE>). The message has the following format:

...No parameters ...

Send
>SXAVE<

The unit immediately returns
>RXAVE<

Ending any current voice call.

6.81 (XAVI) Voice Call Identification switch

Qualifiers: S, R, Q

This command is used to turn ON/OFF the voice call number validation performed by the unit on incoming voice calls. It defaults to 'ON', meaning that the calling-party's number has to be defined on any of the Telephone-type destinations in order for the unit to accept the call⁹. If set to 'OFF' the unit will answer any incoming voice call *regardless of the calling-party's number being set or not* on the Telephone-type destinations' space. The message has the following format:

| A | | | | |
|-------|-------------|--------|-------|--|
| Chars | Item | Format | Value | Meaning |
| 1 | Use Call ID | A | 0-1 | Use '1' for turning on voice-call identification (default value). Use '0' for turning off voice-call identification. |

⁹ The access flag of the *Destination* has to indicate that voice calls' are allowed from that number too.

6.82 (XAVM) Microphone gain

Qualifiers: S, R, Q

This command is used to change the microphone gain for voice calls.

| A | | | | |
|-------|--------------|--------|-----------|--|
| Chars | Item | Format | Value | Meaning |
| 1 | Action/value | A | 0-9, +, - | Gain value (0-9) or gain action: + to increase, - to decrease. Default value is 5. |

Example, to set the microphone gain to 8:

>SXAVM8<

To rise the gain one level:

>SXAVM+<

To lower the gain one level:

>SXAVM-<

6.83 (XAVO) Virtual Odometer

Qualifiers: S, Q, R

Use this message to consult the current value of the Virtual Odometer or to set a new value for it. This message can also be used to configure up to five distance thresholds associated with the L signals. A Control Signal can also be configured, which instructs the Virtual Odometer to only increment its value when the Control Signal is “True”. This message has the following format:

| A [BB] [; CCC] DDDDDDDDD | | | | |
|---------------------------|-----------------|------------|---------|--|
| Chars | Item | Format | Value | Meaning |
| 1 | Action | A | flag | V: Set/Query the Virtual Odometer current value. T: Set/Query a Virtual Odometer threshold. C: Set/Query the Control Signal. |
| 2 | Threshold Index | [BB] | 00-04 | |
| 3 | Control Signal | [; CCC] | flag | IGN: Ignition control. F13: External Power control. IPx: Input control. x = 1 to 4. U: Disable Control Signal. |
| 9 | Value | DDDDDDDDDD | integer | Virtual Odometer value or Threshold value. Use “U” to undefine a Threshold. |

Examples To set the Virtual Odometer with the actual vehicle’s odometer, in this case 10458 meters, use the following message:

```
>SXAVOV000010458<
```

To create a threshold of 500km, use the following message:

```
>SXAVOT00000500000<
```

To delete a threshold use:

```
>SXAVOTU<
```

To use the vehicle’s ignition as Control Signal, use the following message:

```
>SXAVOC;IGN<
```

To disable the use of a Control Signal:

```
>SXAVOCU<
```

6.84 (XAVS) Speaker volume

Qualifiers: S, R, Q

This command is used to change the speaker volume for voice calls.

| A | | | | |
|-------|--------------|--------|-----------|--|
| Chars | Item | Format | Value | Meaning |
| 1 | Action/value | A | 0-9, +, - | Volume value (0-9) or volume action: + to increase, - to decrease. Default value is 5. |

Example, to set the speaker volume to 3:

>SXAVS3<

To rise the volume one level:

>SXAVS+<

To lower the volume one level:

>SXAVS-<

6.85 Errors list

The following table contains a list of the errors returned by the unit with the **ER** message. See the **ER** message for more information.

6.85. ERRORS LIST

| ID | Description |
|----|---|
| 00 | Unrecognized command |
| 02 | The message is not delimited by > and/or < |
| 03 | ID miss match on incoming postfix ";ID= " |
| 04 | KY miss match on incoming postfix ";KY= " |
| 06 | Not a valid set message. |
| 07 | Missing parameter. |
| 08 | TAIP console is restricted for this DP. See the locking parameters on the XADP message. |
| 09 | Queries resulting on multiple answers are not supported over the air. |
| 10 | Invalid Speed Threshold query. See the GS message. |
| 11 | Invalid Time Window query. See the GT message. |
| 12 | Invalid Time and Distance query. See the TD message. |
| 14 | Canned Reply list is full. |
| 15 | Invalid Region query. See the GR message. |
| 16 | Invalid Event query. See the ED message. |
| 17 | Invalid Destination Address query. See the DA message. |
| 18 | Invalid Destination Point query. See the XADP message. |
| 19 | Invalid index. |
| 20 | Message can not be used with D qualifier. See the Introduction Chapter. |
| 21 | SRT;ALL can only be used locally. Use the serial port. |
| 22 | The unit is restarting and it can not process any command. |
| 23 | Minimum time/date can not be ahead of Maximum time/date on a Time Window. Time Window can not be less than 2 minutes |
| 24 | Invalid characters on string parameter. |
| 25 | Invalid vehicle ID. See the ID message. |
| 26 | PIN can only be modified locally. Use the serial port. |
| 27 | APN can not be changed via IP communication. Use serial port or SMS messages. |
| 28 | Unrecognized RF parameter. See the RF message. |
| 29 | RF parameter longitude exceeded. See the RF message. |
| 30 | Set messages are locked for the serial port. See the XALL message. |
| 31 | Wrong unlocking code. See the XALL message. |
| 32 | Invalid MDT PAD Size. See the MT message. |
| 33 | Invalid MT format. See the MT message. |
| 34 | Invalid MDT PAD Timeout. See the MT message. |
| 35 | Event definition space exceeded. See the ED message. Try dividing a long event into several cascading events using <i>event signals</i> and <i>user signals</i> . |
| 36 | <i>Event sense</i> missing on event definition. See the ED message. |
| 37 | Non-existent signal on event definition. See the ED message and Signals Chapter. |
| 38 | PAD1 or PAD2 must differ from PAD escape and PAD entry. See the MT message. |
| 39 | Invalid KY set format. |
| 40 | The signal can not be changed by user. See the SS message. The signal depends on the units' situation and so it can not be arbitrary modified. Only <i>user signals</i> and <i>outputs</i> can be manipulated directly by the user. |
| 41 | An input can not be modified. Only outputs may be modified with the SS message. |
| 42 | Invalid signal index. See the Signals Chapter. |
| 43 | Pending SMS message. Can not send a new SMS message until previous message is sent or canceled. |

The table continues on the next page.

6.85. ERRORS LIST

Continued from the previous page.

| ID | Description |
|----|--|
| 44 | Invalid XAGA or XAGB query. |
| 45 | Wrong XAGA or XAGB parameter. |
| 46 | Internal communication problem. |
| 47 | Length exceeded for TX message. See the TX message. |
| 48 | Invalid event index when using the XATM message. See the XATM message. |
| 49 | Invalid XATM message format. See the XATM message. |
| 50 | Invalid reset option. See the RT message. |
| 51 | Can not modify a DP using the TAIP console from the same DP. |
| 52 | Can not use 0000 as locking code. See the XALL message. |
| 53 | There has to be at least one input to wake up the unit. See the XAPM message. |
| 54 | Invalid parameter on firmware server configuration. |
| 55 | Invalid size on firmware server configuration. |
| 56 | Invalid format in XAFU message. See the XAFU message. |
| 57 | Invalid firmware server. See the XAFU message. |
| 58 | An over-the-air firmware upgrade process is already taking place. |
| 59 | Invalid parameter range/value. |
| 60 | IP-type destination supplied with no port. See the XADP message. |
| 61 | Telephone-type destination supplied can not include port parameter. See the XADP message. |
| 62 | Functionality not available on this product. |
| 63 | Can not initiate a new voice call: There is a voice call being held. |
| 64 | Can not initiate voice call: The selected <i>Destination</i> is not defined. |
| 65 | No current voice call to end. |
| 66 | No password has been set for the authentication mechanism. |
| 67 | Message ID already in use by the Garmin device. |
| 68 | Exceeded the allowed length for this command. |
| 69 | Firmware upgrade over the air is not enabled on this module. |
| 70 | Garmin Mode is not enabled. |
| 71 | The change ID for the Driver Status change already exists. |
| 72 | The Driver Status ID does not exist. |
| 73 | Driver Status list is full. |
| 74 | The Driver Status could not be deleted. It is currently being used. |
| 75 | The Driver ID list is full. |
| 76 | Invalid counter operation. See the GC message. |
| 77 | Invalid Counter recycle flag. See the GC message. |
| 78 | Counter Threshold or Value parameter required. See the GC message. |
| 79 | Counter command not valid for the actual recycle flag. See the GC message. |
| 80 | Driver ID does not exist on the list. |
| 81 | Unrecognized Set message. |
| 82 | Wrong region index. See the GR message. |
| 83 | No space available for cells format. |
| 84 | At least one subscribed cell, can not format. |
| 85 | Found illegal characters on a <i>Destination Point</i> definition. See the XADP message. |
| 86 | Can not change a GPIOs mask (GF) on this product. Only the F0 value is accepted. See the GF message. |

The table continues on the next page.

6.85. *ERRORS LIST*

Continued from the previous page.

| ID | Description |
|----|---|
| 87 | Problem with the Canned Reply ID. Try a new ID. |
| 88 | Wrong point index. See the GR message. |
| 89 | Checksum error. Mismatch or checksum was not present in the response message. |
| 90 | Unavailable feature. |
| 91 | Operation not allowed. Verify that the MDT mode is not active before trying to enable Garmin Mode. |
| 92 | Canned Reply could not be removed. |
| 93 | Invalid Canned Reply count. Send a Canned Reply Text Message using 1 to 6 Canned Replies. |
| 94 | Canned Reply ID does not exist in the list. |
| 95 | Invalid or duplicated Canned Reply ID. |
| 96 | Internal flash memory error. Can not save on flash. |
| 97 | Event code already in use. |
| 98 | Feature not implemented on the Garmin device. |
| 99 | GPS data temporarily unavailable. If this situation persists for more than 30 seconds you may have a hardware problem. It is normal to get this error when a GPS request is issued too soon after power-up. |

7 Appendix A - Quick Start Guide

The next two pages are a quick guide for installing and configuring the unit. You can print these two pages and have them at hand.

- ☐ Antares SB
- ☐ I/O Harness
- ☐ Power Harness
- ☐ GPS Antenna
- ☐ GSM Antenna

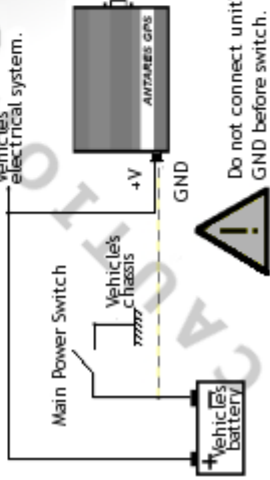
Ignition Not connected



| HARNESS COLOR CODE | |
|--------------------|--------|
| Signal | Color |
| +V | Red |
| GND | Black |
| Ignition | Yellow |

(Antares side)

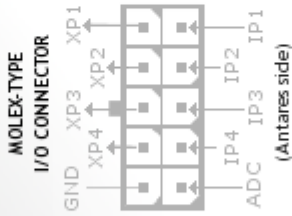
Do not connect unit's GND before switch.

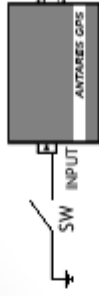


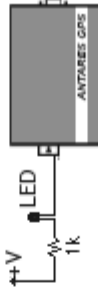
- No contact with water.
- No direct exposure to sun light.
- Away from excessive heat sources like the motor or the exhaust's path.
- Away from excessive cold sources like a truck's refrigerator or AC system.
- Not attached to a highly vibrating structure.

Operating Temperature: -20 °C to +55 °C
-4 °F to +131 °F

| HARNESS COLOR CODE | |
|--------------------|--------------|
| Signal | Color |
| Input 1 IP1 | White/Brown |
| Input 2 IP2 | White/Red |
| Input 3 IP3 | White/Orange |
| Input 4 IP4 | White/Yellow |
| ADC | White |
| Out 1 XP1 | Blue/Brown |
| Out 2 XP2 | Blue/Red |
| Out 3 XP3 | Blue/Orange |
| Out 4 XP4 | Blue/Yellow |
| GND | Black |

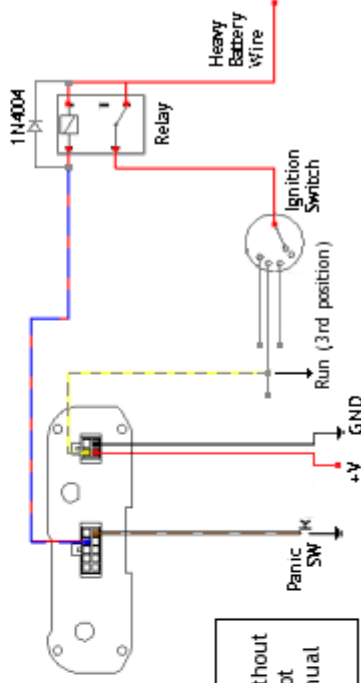


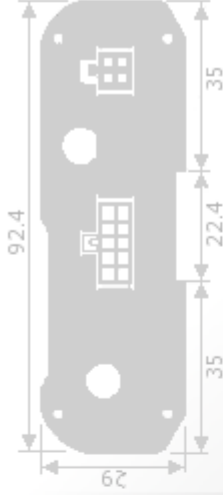




- Panic Switch
- Ignition Sense
- Ignition Drive

CAUTION:
Driving a vehicle's engine off without knowing the vehicle's state is not advised. Refer to the unit's manual for recommendations.





Depth = 105mm

QUICK START GUIDE 2/2

CONFIGURATION

PREREQUISITES

- ☐ 8 to 32V DC 0.8A Power Source.
- ☐ RS-232 or USB/RS-232 cable.
- ☐ Pre-installed Antares Configuration Tool Software Version 1.6.4 or superior. (Free software)
- ☐ GPRS enabled SIM card: PIN and APN required.
- ☐ Getting Started [tmf] script file or User's Manual.
- ☐ An AVL server listening for TAIP reports on TCP.

3. CONNECT AND POWER ON

- a. Connect the GSM/GPRS antenna to the unit.
- b. Connect the RS-232 serial cable to the unit and to the computer.
- c. If inserted, remove the SIM card from the unit.
- d. Having previously attached a power source to the power harness, connect the power harness to the unit.
- e. Wait for the Power LED to stay solid ON. Normally this takes about 15 seconds after power up*.

* The stabilization of the Power LED could take up a minute on a unit with a discharged built-in battery.

6. INSERT SIM CARD

- a. Disconnect the unit's power harness.
- b. Insert the SIM card as indicated on the figure until it clicks.



- c. Connect the power harness to the unit.

Note: When working with a built-in back-up battery ignore steps 'a' and 'c', and reset the unit after 'b' with the >SRT < command.

1. GETTING STARTED SCRIPT

If you already have the [GettingStarted.tmf] script file proceed to the next step.

- a. Open Windows(TM) notepad program or any other plain-text editor and create a new empty file.
- b. Copy the content of the Antares SB User's Manual Getting Started Script Appendix and paste it on the new text file.
- c. Save the file as GettingStarted.tmf or any other convenient name preserving the [tmf] extension.

For notepad users: Make sure [All files] is selected when saving the file or else the file will be saved as [tmf.txt]

4. COMMUNICATE WITH THE UNIT

- a. Close any application that uses serial ports.
- b. If using an USB/RS-232 converter connect it now.
- c. Open ACT software.
- d. On the [Comm Port] menu select the serial port the unit is connected to.
- e. Click the first [Read from Antares] button and wait for the Version an ID fields to be filled.
- f. Confirm that the Version field reads: Antares GPS 05.22

7. CHECKING THE AVL SERVER

- a. If the ON LINE led does not get solid ON after 15 minutes, refer to the User's guide for Troubleshooting.
- b. If the ONLINE led is solid ON, an event report should be arriving to the AVL server on a 5 minutes interval. The report code is 37.
- c. An event report should be arriving also when Input 3 is connected to GND. The report code is 05.

Refer to the User's Manual for more information and troubleshooting.

2. EDIT/ADJUST THE SCRIPT

Open the [GettingStarted.tmf] script file with a text editor like Windows(TM) notepad and:

- a. Change the [EXAMPLE] ID for your ID: >SIDEEXAMPLE <
- b. Change the [1234] SIM card's PIN: >SRFI1234 <
Or use >SRFI < for no-PIN.
- c. Change the [internet.carrier-name.com] APN: >SRFAInternet.carrier-name.com <
- d. Change the [avl.server.com:2145] AVL server: >SXAIP0000avl.server.com:2145 <
- e. Save the file.

5 . PASS THE SCRIPT TO THE UNIT

- a. On ACT's [Antares] menu select [Write configuration to antares from file].
- b. When prompted, select the [GettingStarted.tmf] script file and click the [Open] button.
- c. The writing process starts. Wait for the [Sending Message] dialog to disappear. The Getting Started script is written in less than 5 seconds.

INFORMATION

Antares SB is manufactured by
DIGITAL COMMUNICATIONS TECHNOLOGIES

You can get this Quick Start Guide and the unit's manual from
www.digitalcomtech.com

For technical support contact DCT at:
support@digitalcomtech.com

This guide applies to Firmware version 5.22.
For more information refer to the Antares SB User's Manual.
When using TAIP Downloader make sure of using version 1.0.2 or superior, go to [Help] [Version] to consult.
For instructions on upgrading go to DCT's web site.

8 Appendix B - Getting Started Script

Select the following script and paste it on any plain-text editor to create the *GettingStarted.tmf* script file.

```
#Antares SB script
#Getting Started example

#Delete any previous configuration
>SRT;CONFIG<
>SXADP**U<

#Unit's ID
>SIDEEXAMPLE<

#configuring the SIM's PIN
>SRFI1234<

#configuring the APN
>SRFAinternet.carrier-name.com<

#The remote AVL server address and port
>SXADP0000avl.server.com;2145<

#A Destination Address holding the
# server destination
>SDA4;P00<

#Time-only Time And Distance
#signal definition
>STD80300<

#Event triggered by T&D signal
>SED37NV4;TD8+<

#Input report event
>SED05NV4;IP3+<

#end
```

9 Appendix C - Signals' Table

For more information on signals refer to the *Configuration* chapter, *Event Machine* section. The available signals for the *Antares SBTM* are presented in the next table on the next page.

| ID | Signal | Description |
|-----------|--------------------------------------|---|
| A00 - A09 | <i>Destination Points'</i> state | True when the IP address/port defined on the corresponding <i>Destination Point's</i> index is accepting a TCP connection or when using UDP (i.e. the TCP/UDP socket is open). See the XADP message. |
| B00 - B04 | Battery Levels(+) | True when the unit's back-up battery level is above the value defined with the XAGB message. |
| C00 - C19 | Counters, Timers, Distancers(+) | True when the corresponding counter reaches its defined threshold value. See the GC message. |
| D00 - D04 | ADC Levels(+) | True when the ADC input voltage level is above the value defined with the XAGA message. |
| E00 - E49 | Event Triggers(+) | True when the corresponding event trigger is <i>True</i> . See the ED message. |
| F00 | Ignition | True when the ignition input of the unit is on. |
| F01 | GPS Fix | True when doing GPS fixes. |
| F02 | GSM/GPRS Roaming | True when the unit is Roaming on GSM/GPRS. |
| F03 | GSM-Registered | True when the unit is registered in the GSM network. |
| F04 | GPS Antenna Feed-line fault | Indicates a short on the GPS antenna cable. |
| F05 | GPRS-Registered | True when the unit is <i>registered</i> on the GPRS network. |
| F08 | GPRS-Attached | True when the unit is <i>attached</i> to the GPRS network. |
| F09 | MDT PAD Mode | True when the unit is in MDT PAD mode. See the MT message |
| F10 | PAD message | True when an MDT message is received on the serial port when PAD mode is ON. See the MT message |
| F11 | Woke-Up | True when the unit wakes-up after sleep power mode. Immediately reset after the first events' evaluation. See the XAPM message |
| F12 | 12volts/24volts detector | True when the main supply voltage is above 16 volts. |
| F13 | Power | True when the unit's main power supply is on. |
| F14 | Cell ID change | This signal transitions to True when the unit registers in a new cell. |
| G00 - G07 | General Purpose Input Outputs(*)(-). | These signals are true when the corresponding GPIO is true. See the SS and GF messages. |
| H00 - H04 | Store & Forward Thresholds | True when the ammount of messages stored in the S&F Buffer exceeds the corresponding threshold value. See the XAGF message. |
| IP1 - IP4 | Inputs. | True when the corresponding input is on. See the SS message. |
| J00 - J04 | Heading Deltas. | True when the vehicle's heading change is greater than the corresponding heading change threshold. The signal is immediately reset after evaluation to achieve a turn-by-turn report. See the SXAGH message. |
| K00 - K99 | Circular Regions. | True when the vehicle is inside the corresponding Circular Region. See the SXAGR and XAIR messages. |
| L00 - L04 | Virtual Odometer Thresholds | True when the virtual odometer value exceeds the corresponding threshold value. See the XAVO message. |
| N00 - N04 | Acceleration. | For positive acceleration thresholds: True when the vehicle's acceleration is larger than the corresponding threshold. For negative acceleration thresholds: True when the vehicle's acceleration is less than the corresponding threshold. See the XAGN message. |

| ID | Signal | Description |
|-----------|--------------------------------------|---|
| OE0 - OE4 | Engine's RPM thresholds | True when the engine's RPM detected by the OBD is higher than the corresponding RPM thresholds. See the XA0E message. |
| OT0 - OT4 | Throttle Position thresholds | True when the throttle position percentage is higher than the corresponding throttle thresholds. See the XA0T message. |
| OF0 - OF4 | Fuel Level percentage | True when the remaining fuel in the vehicle drops below the corresponding fuel thresholds. Percentage. See the XA0F message. |
| OG0 - OG4 | Fuel Gallons thresholds | True when the remaining fuel in the vehicle drops below the corresponding fuel thresholds. Gallons. See the XA0G message. |
| OR0 - OR4 | Fuel Rate thresholds | True when the fuel consumption is higher than the corresponding fuel rate thresholds. Gallons per hour. See the XA0R message. |
| OS0 | Status: Communicating with Vehicle | True when the OBD is communicating with Antares. See the XA0S message. |
| OS1 | Status: Communication Checksum error | True when an OBD communication checksum error has occurred. See the XA0S message. |
| OS2 | OBD Vehicle Ignition State | True when the OBD detects that the vehicle ignition is on. See the XA0S message. |
| OS3 | Vehicle Malfunction Indicator Light | True when the OBD detects that the Vehicle Malfunction Indicator Light (MIL)(Check Engine Indicator) is on. See the XA0S message. |
| OS4 | Secondary OBD tool connected | True if there is a secondary OBD is tool connected. See the XA0S message. |
| R00 - R29 | Regions(+) | True when the vehicle is inside the corresponding region. See the GR message |
| RST | Software reset | This signal is used to detect a firmware reset. It is set to true every time there is a software reset and it is immediately reset after the first events' evaluation. |
| S00 - S09 | Speed thresholds(+) | True when the vehicle's speed is faster than the corresponding speed threshold. See the GS message. |
| T00 - T09 | Time Windows(+) | True when the time and date are within the corresponding time window. See the GT message. |
| TD0 - TD9 | Time and Distance(+) | True when the associated <i>Time and Distance</i> counter has a Time and Distance condition true. The signal is immediately reset after being evaluated to enable the counter for further triggers. See the TD message. |
| U00 - U09 | User flags(-) | These signals may be changed by the user at any time with the SS message. |
| V00 - V05 | Voice signals | V00 signal is <i>true</i> when there is a voice call on course. V01 to V05 are true whenever the actual voice call is being held to <i>Destination Point</i> 10 to 14 respectively. See the VC, VE and XADP messages. |
| XP1 - XP4 | Outputs(-) | True when the corresponding output is on. See the SS message. |

(*) *Antares SBTM* does not have GPIOs. These signals are maintained for backwards compatibility with scripts from other versions. Instead, *Antares SBTM* has 4 inputs and 4 outputs. Signals G00-G03 are the same as IP1-IP4 signals and G04-G07 signals are the same as XP1-XP4 signals. The user is encouraged to use IP and XP signals instead of G signals.

(-) These signals' state can be directly altered by the user with the SS TAIP message.

(+) Refer to the next paragraph.

The signals marked with a (+) are signals for which the conditions that set them *true* or *false* are defined by the user. The other signals transitions' criteria can not be altered as they depend on conditions already programmed on the unit.

For more information on signals refer to the *Configuration* chapter, *Event Machine* section.

10 Appendix D - Quick TAIP reference

This chapter presents quick examples of the most commonly used TAIP commands. Refer to the *Unit's TAIP reference* chapter for detailed information on a command.

10.1 Setting the *Antares SBTM* ID

To set the unit's ID to *AbcD-1234*, send:
>SIDAbcD-1234<

10.2 Setting the APN

To set the APN to *internet.carrier.com*, send:
>SRFAinternet.carrier.com<

10.3 Configuring the SIM card PIN

To set the PIN to *1234*, send:
>SRFI1234<

To set the PIN as empty, send:
>SRFI<

10.4 Restarting the unit

To make a software and hardware reset on the unit:
>SRT<

10.5 Restoring to factory defaults

To erase the configuration and restore to factory initial values:
>SRT;ALL<

The unit will make a software reset after restoring the factory defaults.

10.6 Resetting the GPRS connection

To reset the GPRS connection, send:

>SXAGP1<

The unit will close the GPRS session and will automatically start it again after approximately 1 minute.

10.7 Configuring a host address/type

To set IP address *192.168.0.1* working on TCP port *8000* on *Destination Point 00*, send:

>SXADP0001192.168.0.1;8000<

To set the same IP/port on UDP *without* confirmation, send:

>SXADP0002192.168.0.1;8000<

To set the same IP/port on UDP with confirmation, send:

>SXADP0003192.168.0.1;8000<

To set the host name *server.name.com* rather than an IP address.

On TCP:

>SXADP0001server.name.com;8000<

On UDP without confirmation:

>SXADP0002server.name.com;8000<

On UDP with confirmation:

>SXADP0003server.name.com;8000<

To delete host address *00*, send:

>SXADP00U<

To delete telephone *10*, send:

>SXADP10U<

10.8 Configuring a telephone number for SMS and Voice interaction

To set the number *9123456789* on *Destination Point 10* for SMS and Voice interaction, send:

>SXADP10109123456789<

10.9 Querying hosts/ports and telephones

To get a list of hosts, ports with working protocol (TCP, UDP), mode (ACK, no-ACK), and telephones, as well as authorization flags for each host and telephone, send:

>QXADP<

10.10 Grouping AVL servers on *DAs*

To make *Destination Address (DA) 4* represent only *host address (Destination Point) 00*, send:

>SDA4;P00<

To make *Destination Address (DA) 4* represent *Destination Point 00* and unit's Serial Port, send:

>SDA4;P00,P15<

To delete *DA 4* grouping, send:

>SDA4U<

10.11 Defining a periodic timer

There are two alternatives to achieve a periodic timer.

10.11.1 Using a time counter

To create a 5-minutes (300 seconds) periodic counter signal (C08), send:

>SGC08TR00300<

or alternatively, use a counter delta of 60 seconds:

>SGC08TR0000500060<

With the same result.

10.11.2 Using a Time And Distance counter

To create a 5-minutes (300 seconds) periodic TD signal (TD4), send:

>STD40300<

10.12 Creating an event

To define event *33* to trigger with the periodic signal *C08* defined on the previous section.

>SED33XV0;C08+<

The letter *X* on the command indicates to generate event reporting messages only to the serial port.

10.13 Creating a turn-by-turn (heading change) report

To define a heading-delta change of 45 degrees, send:

>SXAGH001045<

To define event *44* so that it sends a message to the serial port every time the

10.14. CREATING A KILOMETER COUNTER

vehicle's heading delta changes by more than 45 degrees, send:
>SED44XV0;J00+<

10.14 Creating a kilometer counter

To make a distance counter count kilometers (01000 meters), send:
>SGC12DC0000101000<

To query the kilometer count of this counter, send:
>QGC12V<

To make the unit report the kilometer count automatically: Define an event and add the **XACR** command to the event's user-action field:
>SED22XV0;F00-;ACT=SXACR12A3<

In this example, the event is triggered by a vehicle's ignition turn off. The reported counter is the number 12 and the *Destination Address* is 3.

10.15 Setting an output

To set output 3 *high*, send:
>SSSXP31<

To set output 2 *low*, send:
>SSSXP20<

10.16 Querying the state of an input

To query the state of input 4, send:
>QSSIP4<

10.17 Querying the state of the vehicle-ignition input

Send:
>QSSF00<

10.18 Querying the Analog to digital converter

To query the voltage value at the ADC input, send:
>QXAAC<

10.19 Querying the Internal back-up battery level

To query the voltage value and charge level of the internal back-up battery, send:
>QXABS<

10.20 Driving the unit to sleep power mode

To activate the power saving mode, send:
>SXAPMA<

To make the unit go to sleep mode immediately without activating the power saving mode (once the unit wakes up, it won't go to sleep again automatically):
>SXAPMI<

To deactivate the power saving mode, send:
>SXAPME<

10.21 Querying the firmware version

Send:
>QVR<

10.22 Activating PAD mode on serial port

Send:
>SMTP<

To get the serial port back to TAIP console mode, after using the previous PAD mode command, send 0x1B (ESC key) over the serial port.