



AT Command Set
TT8750+AT001

TT8750+ AT command Set

Revision 1.16

06/29/2015

Confidential and Proprietary Information – © 2015 Skypatrol, LLC.
Do not duplicate without express permission from Skypatrol, LLC

| | |
|----------------------------|----------------------------------|
| Document Title | TT8750+ Skypatrol AT Command Set |
| Version | 1.16 |
| Date | 06/29/2015 |
| Status | Released |
| Document Control ID | TT8750+AT001 |

General Notes

Skypatrol, LLC offers this information as a service to its customers, to support application and engineering efforts that use the products designed by Skypatrol, LLC. System validation of this product, designed by Skypatrol, LLC within a larger electronic system remains the responsibility of the customer or the customer's system integrator. All specifications supplied herein are subject to change.

Copyright

This document contains proprietary technical information, which is the property of Skypatrol LLC, copying this document, or any portion, without the expressed permission in writing from Skypatrol, LLC and sharing it is forbidden.

Copyright © Skypatrol, LLC. 2015

Table of Contents

| | |
|---|-----------|
| Table of Contents | 3 |
| Revision History | 7 |
| Introduction | 12 |
| Purpose | 12 |
| Scope..... | 12 |
| Reference and Use..... | 12 |
| Command Syntax..... | 12 |
| Standard AT Commands | 14 |
| Audio Commands | 15 |
| Basic Audio Commands..... | 15 |
| AT\$TTMRG Radio Modem-Microphone Receiver Gain | 15 |
| AT\$TTSVL Speaker Volume..... | 16 |
| AT\$TTSDT Side Tone Volume | 16 |
| Advanced Audio Commands | 17 |
| AT\$TTECHO Echo Control..... | 17 |
| AT\$TTNRCFG Noise Reduction Configuration | 18 |
| GPIO Commands | 20 |
| AT\$TTIODB Debounce specified GPIO for the specified amount of time | 20 |
| AT\$TTIOCO(x) GPIO Bit Control..... | 21 |
| IP Router Commands | 23 |
| AT\$TTPASP Enables Serial Port Pass thru Interface | 23 |
| AT\$TTPASPACT Activate PAD mode | 24 |
| ATO Return to PAD mode | 24 |
| AT\$TTNETIP Display Network Assigned IP & DNS | 25 |
| Message Commands | 26 |
| AT\$TTATPW Set authorization for AT commands for serial, SMS and API..... | 26 |
| AT\$TTSRVDST Device Servers..... | 27 |
| AT\$TTDEVID DEVICE ID | 29 |
| AT\$CGPCO GPRS Authentication..... | 29 |
| AT\$TTMSGMASK Messages format | 30 |
| AT\$TTLOGCL Message Log Clear | 31 |
| AT\$TTLOGDMP Dump Messages to Serial Port | 32 |
| AT\$TTLOGRD Read Log Data | 33 |
| AT\$TTSNDMG Send Message..... | 35 |
| AT\$TTSMSDST Destination Address for SMS Messages | 37 |
| AT\$TTWAKEUP Modem to Server Wakeup/Keep Alive | 38 |
| AT%BYCNT Query (or Reset) the Byte Counters (GPRS Only) * | 39 |
| API Commands | 41 |
| TCP API Commands..... | 41 |

| | |
|---|-----------|
| AT\$TTTCPCAPI TCP API Control | 41 |
| AT\$TTTCPCFG TCP API Configuration | 42 |
| AT\$TTTCPRST TCP API Restart | 43 |
| AT\$TTTCPCNT TCP API Statistics* | 44 |
| UDP API Commands | 45 |
| AT\$TTACKCFG Acknowledgment Message Period & Retry Number | 45 |
| PAD Commands | 47 |
| AT\$AUXURTCFG Auxiliary Serial Port Configuration | 47 |
| AT\$TTPASPCFG PAD State Configuration | 48 |
| AT\$TTPASPCMF PAD Command Features | 50 |
| FOTA Commands | 52 |
| AT\$TTSFOTA Upgrade Firmware OTA | 52 |
| FTP Commands | 53 |
| AT\$TTFSA Generic Flash File System Access* | 53 |
| AT\$TTFTPCN Cancels Current Data Operation with FTP Server* | 56 |
| AT\$TTFTPDEF FTP Definition* | 56 |
| AT\$TTFTPCD Change current working directory* | 57 |
| AT\$TTFTPCL Closes the Connection to the FTP Server* | 58 |
| AT\$TTFTPDIR Directory Listing* | 58 |
| AT\$TTFTPRT Retrieve a Remote File* | 59 |
| AT\$TTFTPOP Opens FTP Connection* | 60 |
| AT\$TTFTPSTR Store a Local File on the FTP Server* | 61 |
| AT\$TTFTPRST Restart the Last FTP Retrieve Operation* | 62 |
| Real-Time Clock Commands..... | 64 |
| AT\$TTRTCAL Real Time Clock Alarm | 64 |
| AT\$TTRTCTI Real Time Clock | 65 |
| AT\$TTDOW Day of the Week | 67 |
| Network Commands | 69 |
| AT\$TTARG Auto Registration | 69 |
| AT\$TTTOLST Tower List* | 70 |
| AT\$TTNETWD Network Watchdog | 71 |
| AT\$TTGSMJDC Jamming Detection | 73 |
| Function Commands | 75 |
| AT\$TTCAIDLST Call ID List Definition | 75 |
| AT\$TTFNT User Defined Input/Output | 76 |
| AT\$TTFNTDEL Delete Function..... | 78 |
| AT\$TTFNTDELA Delete Function (All) | 79 |
| AT\$TTFNTDELR Delete a Range of Function Groups | 79 |
| AT\$TTFNTGQRY Function Group Query | 80 |
| AT\$TTTRGEV Trigger (Test) Input Function..... | 81 |
| AT\$TTCNT User Defined Input Function Counter | 81 |
| AT\$TTCNTQRY Query Counter | 83 |

| | |
|---|------------|
| AT\$TTFNTQRY Query the State or Value of the Specified Input Function | 84 |
| AT\$TTSTOCMD Store AT Command Functions | 85 |
| Miscellaneous Commands | 87 |
| AT\$TTUSRVAL User Values definition..... | 87 |
| AT\$TTICCID Integrated Circuit Card ID | 87 |
| AT%DM Debug Mode* | 88 |
| AT\$TTDEVSTAT Query Device Status* | 90 |
| AT\$TTSRN Serial Number..... | 93 |
| AT\$RESET Reset device | 93 |
| AT\$TTPWSRV Password for Server and APN | 94 |
| AT\$TTMAPLINK Google Map Link..... | 95 |
| AT\$TTMAPLINKACT Control Map Link function | 96 |
| AT\$TTFNTEXE Execute Output Function..... | 96 |
| AT\$TTRSTCFG Reset Method Configuration | 97 |
| AT+ESMLCK SIM Lock Cell Carrier Registration..... | 98 |
| AT+CLCK Facility Lock..... | 99 |
| GPS Commands..... | 101 |
| AT\$TTGPSSTT GPS Command..... | 101 |
| AT\$TTGPSRT - Configure Sending of GPS Data to the Serial Port..... | 102 |
| AT\$TTGPSQRY Read Current GPS ASCII data | 103 |
| AT\$TTGPSRI Set and Query GPS Reset Interval | 104 |
| AT\$TTTODOM GPS Trip Odometer..... | 105 |
| AT\$TTTODOM Virtual Trip Odometer | 106 |
| AT\$TTODOENA Enable odometer with ignition OFF | 107 |
| AT\$TTOVSDEF Define and Query GPS Over speed Interval..... | 107 |
| AT\$TTGEOFNC Geo Fencing a Circular Area | 109 |
| AT\$TTGFIND Circular Geofence Query | 109 |
| AT\$TTGFDB Set Geofence Debounce Count | 110 |
| AT\$TTGFDEL Delete a Range of Geo-Fences..... | 111 |
| AT\$TTPLYGFN Geo Fencing a Polygonal Area | 112 |
| AT\$TTPLYGFIND Polygonal Geo Fencing Query | 113 |
| AT\$TTPLYDEL Delete Polygonal Geofence | 114 |
| AT\$TTROUTE Geo Fencing a Route | 115 |
| AT\$TTROUTEIND ROUTE Geo Fencing Query | 116 |
| AT\$TTROUTEDEL Delete Route Geofence | 117 |
| AT\$TTGPSHEAD GPS Heading Change..... | 118 |
| Motion Commands | 119 |
| AT\$TTACCAM Motion Configuration..... | 119 |
| Buzzer Commands | 121 |
| AT\$TTBUZCFG Buzzer Configuration | 121 |
| AT\$TTBUZCTR Buzzer Control | 122 |
| Macros | 123 |

| | |
|--|------------|
| AT\$TTSP Set protocol..... | 123 |
| AT\$TTDI Set Distance Interval..... | 124 |
| AT\$TTDL Dump Log Records..... | 124 |
| AT\$TTHB Set Heartbeat Period..... | 125 |
| AT\$TTHC Set Heading Change Warning..... | 126 |
| AT\$TTIPHB Set Heartbeat when IP Change..... | 126 |
| AT\$TTIW Set Idle Warning..... | 127 |
| AT\$TTNR Set Network Restart Period..... | 128 |
| AT\$TTRI Set Report Interval..... | 128 |
| AT\$TTRN Report Current Position..... | 129 |
| AT\$TTRR Set Report After Reset..... | 130 |
| AT\$TTSI Set Interrupt..... | 130 |
| AT\$TTSR Set output state..... | 132 |
| AT\$TTTA Set Tow Alert Warning..... | 132 |
| AT\$TTTE Set Trip event..... | 133 |
| AT\$TTRBI Set Report Interval based on ignition..... | 134 |
| Groups and counters..... | 135 |
| Function Tables..... | 137 |
| Function Type..... | 137 |
| Function Category..... | 139 |
| Input Function Table..... | 140 |
| Output Function Table..... | 144 |
| User Variable Index Table..... | 149 |
| GPIO Flash Table..... | 154 |
| Message Type Table..... | 154 |
| Keep Alive Message..... | 155 |
| Position Report Messages..... | 156 |
| Bit 26 Binary Format Table..... | 159 |
| Bit 26 ASCII Format Table..... | 160 |
| Counter messages..... | 162 |
| Variable Message..... | 165 |
| Geofence Message..... | 168 |
| Appendix A – Result Codes..... | 172 |
| Result Codes..... | 172 |
| Unsolicited Result Codes..... | 172 |
| SMS Unsolicited Result Codes..... | 173 |
| Appendix B – Error Codes..... | 174 |
| CME Error Codes..... | 174 |
| CMS ERROR Codes..... | 176 |
| Summary of URC..... | 177 |
| Index..... | 181 |

Revision History

| Version | Primary Author(s) | Description of Version | Date Completed |
|----------|--------------------------------|---|----------------|
| Rev 1.00 | Erik Ordonez Ramiro Sanchez | Initial revision. | 08/12/2011 |
| Rev 1.01 | Erik Ordonez | | 09/12/2011 |
| Rev 1.02 | Erik Ordonez Ramiro Sanchez | <ul style="list-style-type: none"> Position Message Format Byte modification stating at 21 to 26 TTCNYQRY, types definition updated Add new parameter to AT\$TTRTCAL <rtc_index> Add the new command AT\$TTDOW Change param1 values in the event category 28 Add the parameter <index> to command AT\$TTTRGV Change the values of param1 to 0-1200 for type of function 16 Modify the index for the command AT\$TTOVSDEF, (only 3 over speed definition are available) Modify the AT\$TTROUTE command (index will be from 1 to 25) Modify the AT\$TTROUTEDEL and AT\$TTROUTEIND commands Modify the AT\$TTPLYGFN , AT\$TTPLYGFIND and AT\$TTPLYDEL commands Include definition of the AT%DM command Add more bits to setup the report mask and sequence number for each type of report (Keep alive message, Position report messages, Counter messages, Variables messages, Geofence messages) | 12/12/2011 |
| Rev 1.03 | Ramiro Sanchez | <ul style="list-style-type: none"> Add new output event increment counter (category 53) Add new output event decrement counter (category 54) Add new output event enabled counter (category 55) Add new output event disabled counter (category 56) Add the parameter <group> for the command AT\$TTCNT. Change <param1> definition for output event category 20,21,22,23,24,25 and 26 | 12/14/2011 |

| Version | Primary Author(s) | Description of Version | Date Completed |
|----------|---|--|----------------|
| Rev 1.04 | Erik Ordonez | <ul style="list-style-type: none"> • Update the number of I/O to 9 in the AT\$TTIOCO# • Update Index ID for AT\$TTMSGMASK • Add a new type and update queue type definition for AT\$TTLOGCL • Add a new type and update queue type definition for AT\$TTLOGDMP • Add a new type and update the queue type definition for AT\$TTLOGRD • Remove the <gateway number> param for usage of email in the AT\$TTSMSDST | 01/18/2012 |
| | Erik Ordonez | <ul style="list-style-type: none"> • Update label for AT\$TTTRGV • Remove the number of Indexes for AT\$TTRTCAL • Update functionality for AT\$TTGFIND, AT\$TTPLYGFN, AT\$TTROUTE • Add the new commands to query all geofence definitions AT\$TTGFIND, AT\$TTPLYGFIND, AT\$TTROUTEIND. • Add new command AT\$TTGSMJDC • Replace and update command definition AT\$TTACCAM with AT\$TTMOTCFG • Update Input & Output Function table • User Variable Index Table used only with Output Function | 01/19/2012 |
| Rev 1.05 | Flor Hernandez Camilo Moreno Ramiro Sanchez | <ul style="list-style-type: none"> • Remove extra parameters in the command AT\$TTGPSRI • Change the limit for Odometer (new value: 2147483647) • Change the limit for Trip Odometer (new value: 2147483647) • Add parameter <new odometer> to the command AT\$TTTODOM | 01/31/2012 |
| Rev 1.06 | Flor Hernandez Camilo Moreno Ramiro Sanchez | <ul style="list-style-type: none"> • Command AT\$TTIOPULDN was removed | 02/01/2012 |

| Version | Primary Author(s) | Description of Version | Date Completed |
|----------|-------------------|--|----------------|
| Rev 1.07 | Ramiro Sanchez | <ul style="list-style-type: none"> Remove input function category 4 (Input 3) Remove input function category 2 (output 1) Remove input function category 5 (output 2) Remove input function category 6 (output 3) Remove input function category 7 (USR LED 1) Remove input function category 8 (USR LED 2) | 02/03/2012 |
| Rev 1.08 | Erik Ordonez | <ul style="list-style-type: none"> Add Buzzer commands AT\$TTUSRVAL | 03/06/2012 |
| Rev 1.09 | Ramiro Sanchez | <ul style="list-style-type: none"> Remove Description Section Remove TTFOTAUG command Remove TTUGST command Remove vale 3 from <host interface> in the command AT\$TTPASP Add command AT\$TTSRN Add command AT\$RESET Replace and update command definition AT\$TTMOTCFG with AT\$TTACCAM Remove Extended Error code | 03/30/2012 |
| Rev 1.10 | Ramiro Sanchez | <ul style="list-style-type: none"> Mark commands do not supported | 04/17/2012 |

| Version | Primary Author(s) | Description of Version | Date Completed |
|----------|-------------------|---|----------------|
| Rev 1.11 | Ramiro Sanchez | <ul style="list-style-type: none"> • Modify range of <rxgain> in the command AT\$TTMRG • Remove the definition of the command AT\$TTSTG Speaker Transmit Gain • Modify range of <volume> in the command TTSVL • Remove the definition of the command TTMICE Echo Cancellation • Remove the definition of command AT\$TTMINR Ambient Noise Reduction Control • Add command AT\$TTNRCFG Noise Reduction Configuration • Add command AT\$TTECHO Echo control • Add command AT\$TTSdT Side Tone Volume • Remove the definition of the command AT\$TTMIES Echo Suppression Control • Remove the definition of the command AT\$TTPAMP Set Uplink Voice Parameters • Remove the definition of the command AT\$TTSPCFG Set Downlink Voice Parameters • Remove the definition of the command AT\$TTVSEL Voice Select • Add command TTMAPLINK Google Map Link • Change general error codes • Remove SAT Application Toolkit Result Codes • Add command AUXUARTCFG Auxiliary Serial Port Configuration • Add parameter <uart> to the command AT\$TTPASP • Add command AT\$TTPWSRV Password for Server and APN • Add parameter <flash> to the command AT\$TTSNDMG • Add output function 57 and 58 | 06/29/2012 |
| Rev 1.12 | Ramiro Sanchez | <ul style="list-style-type: none"> • Add command AT\$TTFNTEXE to execute an output function. • Add Macro definitions • Modify limit of param1 and param2 for categories 16 and 17 • Modify limit of param1 in category 21 | 10/16/2012 |

| Version | Primary Author(s) | Description of Version | Date Completed |
|----------|-------------------|---|----------------|
| Rev 1.13 | Ramiro Sanchez | <ul style="list-style-type: none"> • Add parameter <port need> to the command AT\$TTACKCFG • Add parameter <show ok> to the command AT\$TTFNTEXE • Add parameter <period> to the command AT\$TTGSMJDC • Add command AT\$TTIW to setup the idle warning • Add AT\$TTRR commando to setup report after reset • Add AT\$TTA command to setup the tow alert warning • Add AT\$TTRBI command to setup the periodic report base in ignition • Add AT\$TTGPSHEAD command to set an angle threshold to monitor GPS heading change | 11/08/12 |
| Rev 1.14 | Ramiro Sanchez | <ul style="list-style-type: none"> • Add input function category 51, heading change | 11/16/12 |
| Rev 1.15 | Renato Motta | <ul style="list-style-type: none"> • Changed AT\$TTTA Definition • Removed duplicate of AT\$TTTA • Added more detailed description of features and commands. • Updated Message mask table bit 2 on ASCII Messages • Updated Message mask table bit 22 size of RTC field on ASCII messages. • Added command AT\$TTMAPLINKACT • Added command AT\$TTRSTCFG • Added command AT+ESMLCK • Added command AT+CLCK • Fixed description of Output events 44,45 and 46 | 03/11/2014 |
| Rev 1.16 | Ramiro Sanchez | <ul style="list-style-type: none"> • Added command AT\$TTODOENA to enable or disable odometer with ignition OFF • Added input function 64 to detect acceleration based on GPS • Added input function 82 to detect GSM signal quality • Added new value for the parameter <destination> in the command AT\$TTSNDMG | 05/28/2015 |

Introduction

Purpose

Scope

The Air Interface Protocol is a digital communication interface over SMS or GPRS which is used for all communication between the backend server and the terminal. The backend server sends a command to the terminal and then the terminal confirms with an acknowledgement message. If necessary, the terminal also sends report messages to the backend server.

The purpose of this document is to describe how to build up the backend server based on the Air Interface Protocol; it includes the definition of the functions, commands to setup the basic parameters, definition to the functions and action to be handling by the device.

All commands will be detailed defined and it can be use as a AT command reference manual.

Reference and Use

The TT8750+ will be referred to using various terms, to include: MS (Mobile Station), TA (Terminal Adapter), DCE (Data Communication Equipment), or ME (Mobile Equipment).

The TT8750+ can be controlled via the use of a DTE (Data Terminal Equipment) platform by issuing the AT commands via a serial interface.

Command Syntax

The attention or "AT" prefix is required prior to entering any command. All commands require a carriage return or <CR> following the entry of the desired command. All command responses are encapsulated by a carriage return and line feed or <CR><LF>. The ASCII display of these characters is suppressed with only the modem response being presented.

In addition to terminating AT commands, the carriage return <CR> is also used to abort commands that are executing.

Most AT commands complete immediately so there is no opportunity to abort them, for instance ATI. However, some commands like AT+COPS or AT+CFUN can actually take several seconds to complete. The AT command interface is said to be in execution mode when a command is running and has not returned a result code (OK/ERROR). A second <CR> entered while the AT command interface is in execution mode will abort the command and return the interface to command mode.

Some AT commands require additional input, for instance AT+CMGS. After terminating the AT+CMGS command with a <CR> the AT command interface enters line edit mode. While in line edit mode all characters are accepted except CNTL-Z. CNTL-Z terminates line edit mode and the AT command interface enters execution mode. Like before, at this point another <CR> will abort the command.

AT message concatenation can be done using the “;” <semicolon> between commands. Except commands start with &, in this case is not necessary include the “;”.

The following examples demonstrate the potential usage of AT commands presented:

| Type | Example | Description |
|-----------------------|---------------------------|---|
| Command Format Query | AT+TTXXX=? | When entered will return the command format and value ranges. |
| Command Read | AT+TTXXX? | When entered will return the current value assigned to the command. |
| Command Write | AT+TTXXX=<value>,<value>, | When entered will set the command to specified value(s). |
| Command Execution | AT+TTXXX | When entered will execute the specified command. |
| Command Concatenation | AT+CRC=1;S0=1 | When entered it will execute both the CRC and S0 command. |

Standard AT Commands

The following is the format in which all commands will be presented.

| | |
|----------------------|--|
| AT\$TTXxxx(Command) | Xxxx (Command Description) |
| Command Function | (Description of the command function) |
| Command Format Query | AT\$TT x=? |
| Response | AT\$TT x: (parameter1 name 1 – 15), (parameter2 name 1-10),... |
| Write Format | AT\$TT x=<value>,<value>[,<optional value>],... |
| Response | OK or ERROR |
| Read Format | AT\$TT x? |
| Response | <value>,<value>,... |
| Execution Format | AT\$TT x |
| Response | OK, ERROR, or <value> |
| Parameter Values | |
| <Value 1> | AT\$TT x: (1-15),(1-10) |
| <Value 2> | |
| Notes | (Additional command notes) |
| Examples | |



Note: Where applicable, the <value> responses provided for the READ and EXECUTION formats are modem default values. All efforts will be made by Skypatrol, LLC. to keep these values current in the documentation but will not be responsible for any differences that may occur as a result subsequent software builds and version enhancements.

Audio Commands

Basic Audio Commands

AT\$TTMRG Radio Modem-Microphone Receiver Gain

| | |
|----------------------|---|
| AT\$TTMRG | Microphone Receiver Gain |
| Command Function | This command sets the receiver level gain for the microphone input. |
| Command Format Query | AT\$TTMRG=? |
| Response | \$VGR: (0-31) OK |
| Write Format | AT\$TTMRG=<rxgain> |
| Response | \$TTMRG: <rxgain> OK |
| Read Format | AT\$TTMRG? |
| Response | \$TTMRG: <rxgain> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <rxgain> | 0-12 dB 1-11 dB 2-10 dB 31+12 dB |
| Notes | Receiver gain settings are in 1 dB steps from -12 to +12 dB. |

AT\$TTSVL Speaker Volume

| | |
|----------------------|---|
| AT\$TTSVL | Speaker Volume |
| Command Function | This command is used to set the speaker volume |
| Command Format Query | AT\$TTSVL=? |
| Response | \$SVL: (0-15) OK |
| Write Format | AT\$TTSVL=<volume> |
| Response | OK |
| Read Format | AT\$TTSVL? |
| Response | \$TTSVL: <volume> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <volume> | 0 = Mute 1 = -24 dB 2 = -18 dB 15 = 0 dB |
| Notes | |

AT\$TTSDT Side Tone Volume

| | |
|----------------------|---|
| AT\$TTSDT | Side tone volume |
| Command Function | This command is used to set the volume of the side tone |
| Command Format Query | AT\$TTSDT=? |
| Response | \$SVL: (0-15) OK |
| Write Format | AT\$TTSDT=<volume> |
| Response | OK |

| | |
|------------------|---|
| Read Format | AT\$TTSDDT? |
| Response | \$TTSDDT: <volume> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <volume> | 0 = Mute 1 = -24 dB 2 = -18 dB 15 = 0 dB |
| Notes | |

Advanced Audio Commands

AT\$TTECHO Echo Control

| | |
|------------------|--|
| AT\$TTECHO | Echo Control |
| Command Function | This command allows the user to configure the echo control settings for the current voice mode |
| Query Format | AT\$TTECHO=? |
| Response | \$TTECHO: (0-3), (0,65535), (0,2048),(0,65355) OK |
| Write Format | AT\$TTECHO=<echo flag>,<control word>,<nlp>, <suppression value> |
| Response | OK |
| Read Format | AT\$TTECHO? |
| Response | \$TTECHO:<echo flag>,<control word>,<nlp>, <suppression value> OK |
| Execution Format | N/A |

| | |
|---------------------|---|
| Response | N/A |
| Parameter Values | |
| <echo flag> | 0 = disable all echo algorithm 1 = enable AEC (Acoustic Echo Cancelation) echo algorithm 2 = enable EEC (Enhanced Echo Suppression) echo algorithm 3 = enable ES (Echo Suppression) echo algorithm |
| <control word> | 0 = disable all echo algorithm 221 = suitable for handset and headset application 224 = suitable for hand free application |
| <nlp> | 0 = disable the NLP algorithm >0 = enable the NLP algorithm (with greater value, more reduction of echo) |
| <suppression value> | 0 = disable echo suppression algorithm >0 = enable echo suppression algorithm (with smaller value, more reduction of echo) |
| Notes | N/A |

AT\$TTNRCFG Noise Reduction Configuration

| | |
|------------------|---|
| AT\$TTNRCFG | Noise Reduction Configuration |
| Command Function | This command allows the user to configure the noise reduction settings for the current voice mode |
| Query Format | AT\$TTNRCFG=? |
| Response | \$TTNRCFG: (0-1), (0-1), (0-65535) OK |
| Write Format | AT\$TTNRCFG=<ul nr flag>, <dl nr fflag>,<nr level> |
| Response | OK |
| Read Format | AT\$TTNRCFG? |
| Response | \$TTNRCFG: =<ul nr flag>, <dl nr fflag>,<nr level> OK |
| Execution Format | N/A |
| Response | N/A |

| Parameter Values | |
|------------------|--|
| <ul nr flag> | 0 = disable uplink noise reduction controller. 1 = enable uplink noise reduction controller. |
| <dl nr flag> | 0 = disable downlink noise reduction controller. 1 = enable downlink noise reduction controller. |
| <nr level> | Noise reduction controller. 849 = Suitable for handset and headset applications. 374 = Suitable for hand free applications |
| Notes | The Noise reduction controller should not be set to 0 |

GPIO Commands

AT\$TTIODB Debounce specified GPIO for the specified amount of time

| | |
|--------------------|---|
| AT\$TTIODB | De-bounce specified GPIO for the specified amount of time |
| Command Function | This command allows a user to set and query GPIO debounce time. The GPIO must be in the same state for the specified number of seconds before the input function will be triggered. |
| Query Format | AT\$TTIODB=? |
| Response | \$TTIODB: (1-9),(0-60) OK |
| Write Format | AT\$TTIODB=<gpio_number>,<debounce_timeout> |
| Response | OK |
| Read Format | AT\$TTIODB? |
| Response | \$TTIODB: <gpio1>,<gpio2>,<gpio3>,<gpio4>,<gpio5>,<gpio6>,<gpio7>,<gpio8>,<gpio9> OK |
| Execution Format | AT\$TTIODB |
| Response | OK |
| Parameter Values | |
| <gpio_number> | Number of GPIO whose debounce timeout is being set. |
| <debounce_timeout> | Number of consecutive seconds <gpio_number> must be in the same state before its input function will be triggered. |
| <gpio1> | Debounce timeout for GPIO1. |
| <gpio2> | Debounce timeout for GPIO2. |
| <gpio3> | Debounce timeout for GPIO3. |
| <gpio4> | Debounce timeout for GPIO4. |

| | |
|---------|--|
| <gpio5> | Debounce timeout for GPIO5. |
| <gpio6> | Debounce timeout for GPIO6. |
| <gpio7> | Debounce timeout for GPIO7. |
| <gpio8> | Debounce timeout for GPIO8. |
| <gpio9> | Debounce timeout for GPIO9. |
| Notes | If <debounce_timeout> is set to zero, <gpio_number> will not be debounced. <debounce_timeout> is applicable only for <gpio_number> defined as input |

AT\$TTIOCO(x) GPIO Bit Control

| | |
|----------------------|---|
| AT\$TTIOCO(x) | GPIO Bit Control |
| Command Function: | This command allows the user to set the state of the specified GPIO. This command is applicable only for GPIO defined as output |
| Command Format Query | AT\$TTIOCO(x)=? |
| Response | \$TTIOCO(x): (0-1) OK |
| Write Format | AT\$TTIOCO(x)=<mode> |
| Response | OK |
| Read Format | AT\$TTIOCO(x)? |
| Response | \$TTIOCO(x): <current setting> <configured setting> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| (x) | 1-9 GPIO bit |
| <mode> | 0 = off 1 = on |

| | |
|----------|--------------------------------|
| Notes | None |
| Example: | Output format for read command |

AT\$TTIOCO1?

\$TTIOCO1 =1,0

OK

└─── Output pin was configured off
└─── Output pin is currently on

IP Router Commands

AT\$TTPASP Enables Serial Port Pass thru Interface

| | |
|------------------|---|
| AT\$TTPASP | Enable Serial Port Pass Thru |
| Command Function | This command allows the user to configure the desired serial port to modem interface. |
| Query Format | AT\$TTPASP=? |
| Response | \$TTPASP: (0-2), (1-2) |
| Write Format | AT\$TTPASP=<host interface>,<uart> |
| Response | OK |
| Read Format | AT\$TTPASP=? |
| Response | TTPASP: <host interface>, <uart> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <host interface> | 1 = Establish UDP PAD session. Upon establishment of a network activation, a CONNECT message will be displayed. "No Carrier" or error will indicate failed or terminated UDP PAD session. 2 = Establish TCP PAD session Upon establishment of a network activation, a CONNECT message will be displayed. "No Carrier" or error will indicate failed or terminated TCP PAD session. |
| <uart> | A numeric to indicate witch UART will be used 1 = Main serial port 2 = Auxiliary serial port |
| Notes | The serial port will not respond to AT commands if the modem establishes a connect state before the baud rate has been |

| | |
|--|---------------------------------|
| | determined for the serial port. |
|--|---------------------------------|

AT\$TTPASPACT Activate PAD mode

| | |
|------------------|---|
| AT\$TTPASPACT | Enable Serial Port Pass Thru |
| Command Function | This command allows the user to configure the desired serial port to modem interface. |
| Query Format | AT\$TTPASPACT=? |
| Response | \$TTPASP: (0-1) |
| Write Format | AT\$TTPASPACT=<state> |
| Response | OK |
| Read Format | AT\$TTPASPACT=? |
| Response | TTPASPACT: <state> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <state> | 0 = Deactivate PAD mode. 1 = Activate PAD mode. |
| Notes | |

ATO Return to PAD mode

| | |
|------------------|--|
| AT\$TTPASPACT | Return to PAD mode form AT mode |
| Command Function | This command allows the user to return to PAD mode from AT mode in the serial port |
| Query Format | N/A |
| Response | N/A |
| Write Format | N/A |
| Response | N/A |

| | |
|------------------|--|
| Read Format | N/A |
| Response | N/A |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| Notes | After the serial port exits from PAD mode by the sequence characters "+++", AT0 command can let the serial port return to PAD mode if the connection still exists. |

AT\$TTNETIP Display Network Assigned IP & DNS

| | |
|------------------|--|
| AT\$TTNETIP | Display Network Assigned IP & DNS |
| Command Function | This command allows the user to query the modem's network assigned IP. |
| Query Format | N/A |
| Response | N/A |
| Write Format | N/A |
| Response | N/A |
| Read Format | AT\$TTNETIP? |
| Response | <"IP">,<"DNS1">,<"DNS2"> |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <IP> | network assigned IP |
| <DNS1> | network assigned DNS1 |
| <DNS2> | network assigned DNS2 |
| Notes | N/A |

Message Commands

AT\$TTATPW Set authorization for AT commands for serial, SMS and API

| | |
|------------------------------|---|
| AT\$TTATPW | Set authorization for AT commands for serial, SMS and API |
| Command Function | This command allows the user to enable or disable authorization to enter AT commands for the serial, SMS and API. It also sets the password required to run this command |
| Query Format | AT\$TTATPW=? |
| Response | AT\$TTATPW: ("oldpasswd",mask "newpasswd") OK |
| Write Format | AT\$TTATPW= <"oldpasswd", "newpasswd"> |
| Response | OK |
| Read Format | N/A |
| Response | N/A |
| Execution Format Response | AT\$TTATPW=<"passwd",mask> |
| Response | N/A |
| Parameter Values | |
| <oldpasswd> | Specified when the password is being changed. |
| <newpasswd | This is the value of the new password and is specified only when the password is being changed. It must be no more than eight characters in length and must be enclosed in double quotes. |
| <passwd> | Specified when changing the AT command authorization mask. |
| <mask> | Bit mask specifying which interfaces will be authorized to enter AT commands. Each bit specifies one interface as enumerated in the table below. |

| | <table border="1"> <thead> <tr> <th>Bit value</th> <th>Interface</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Serial Port</td> </tr> <tr> <td>2</td> <td>SMS</td> </tr> <tr> <td>4</td> <td>API</td> </tr> </tbody> </table> <p>To select multiple items to authorize, add the bit values of each interface to be authorized. To authorize API and SMS only, the mask value is 6 (4 + 2). AT commands entered over the serial port will not execute and will reply with ERROR.</p> | Bit value | Interface | 1 | Serial Port | 2 | SMS | 4 | API |
|-----------|---|-----------|-----------|---|-------------|---|-----|---|-----|
| Bit value | Interface | | | | | | | | |
| 1 | Serial Port | | | | | | | | |
| 2 | SMS | | | | | | | | |
| 4 | API | | | | | | | | |
| Notes | N/A | | | | | | | | |
| Example: | To set a password on a device that previously had not set \$TTATPW enter: AT\$TTATPW="", "1111" To change a previously set \$TTATPW password enter: AT\$TTATPW="1111", "2222" | | | | | | | | |

AT\$TTSRVDST Device Servers

| | |
|------------------|--|
| AT\$TTSRVDST | Device Server Destination |
| Command Function | This command allows the user to set the device server list. A destination is always allowed remote API access. Destination servers can be configured to receive WAKEUP messages whenever the device receives a new IP, or after a certain period has elapsed. (see AT\$TTWAKEUP) |
| Query Format | AT\$TTSRVDST=? |
| Response | \$TTSRVDST: (1-5),(0,1),"(0-255).(0-255).(0-255).(0-255)",(0-65535),(0-3) OK |
| Write Format | AT\$TTSRVDST =<destination number>, <server indication>,"<destination IP> or <DNS name>",<destination port>, <usage>[,<pwsrv>] |
| Response | OK |
| Read Format | AT\$TTSRVDST? |
| Response | \$TTSRVDST: 01, <server indication>,"<destination IP> or <DNS name>",<destination port>, <usage> |

| | |
|---------------------|---|
| | <p>\$TTSRVDST: 02, <server indication>,"<destination IP> or <DNS name>",<destination port>, <usage></p> <p>\$TTSRVDST: 03, <server indication>,"<destination IP> or <DNS name>",<destination port>, <usage></p> <p>\$TTSRVDST: 04, <server indication>,"<destination IP> or <DNS name>",<destination port>, <usage></p> <p>\$TTSRVDST: 05, <server indication>,"<destination IP> or <DNS name>",<destination port>, <usage></p> <p>OK</p> |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <server number> | Server identification (1-5). |
| <server indication> | 0 = Server is disabled. 1 = Server is enabled. |
| <destination IP> | Server IP value. |
| OR | |
| <DNS name> | Server DNS name |
| <destination port> | Server destination port. |
| <usage> | 0 = Unspecified (treated as UDPAPI) 1 = TCPAPI 2 = UDPAPI 3 = TCPAPI and/or UDPAPI |
| <pwsrv> | Password to modified the destination server, This is an optional parameter and it should be match with the password setup with the command AT\$TTPWSRV |
| Notes | <p>If destination port and usage are not present in the command, it is assumed to be a UDPAPI destination with the default port number.</p> <p>You will use either the Destination IP address or the Destination DNS name, but not both.</p> |

AT\$TTDEVID DEVICE ID

| | |
|------------------|---|
| AT\$TTDEVID | DEVICE ID |
| Command Function | This command allows the user to query/set the device identification. The device id is copied into each wakeup message sent from the modem. (see AT\$TTWAKEUP) |
| Query Format | AT\$TTDEVID=? |
| Response | \$TTDEVID: ("device id ") OK |
| Write Format | AT\$TTDEVID="<device id>" |
| Response | OK |
| Read Format | AT\$TTDEVID? |
| Response | \$TTDEVID: "<device id>" |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <device id> | 0-20 character string in ASCII format. |
| Notes | Default device ID is the IMEI number |

AT\$CGPCO GPRS Authentication

| | |
|------------------|---|
| AT\$CGPCO | GRPS Authentication |
| Command Function | This command allows the user to query/set the user and password for GPRS context activation |
| Query Format | AT\$CGPCO=? |
| Response | \$CGPCO: (1),(0-63),(0,2) OK |
| Write Format | AT\$CGPCO =<Input format>,"<Authentication data>",<cid> |
| Response | OK |
| Read Format | AT\$CGPCO? |

| | |
|-----------------------|---|
| Response | \$ CGPCO: <Input format>,"<Authentication data>",<cid> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <input format> | Format of the password. Always 1 (ASCII) |
| <Authentication data> | Authentication data (ASCII) <username>,<password> where Username: Maximum 64 bytes ASCII string. Password: Maximum 64 bytes ASCII string. |
| <cid> | 0 = The username and password is to be applied to all context Activation. 1 =The username and password is to be applied to Context identifier 1. |
| Notes | Username and Password are case sensitive. |

AT\$TTMSGMASK Messages format

| | |
|------------------|--|
| AT\$TTMSGMASK | Set Message Mask |
| Command Function | This command allows the user to query/set the format of the messages generated by the device. See Message type table for details |
| Query Format | AT\$TTMSGMASK=? |
| Response | \$TTMSGMASK: (0-4),(0-4294967295) OK |
| Write Format | AT\$TTMSGMASK =<Message type>, <message mask> |
| Response | OK |
| Read Format | AT\$TTMSGMASK? |
| Response | \$TTMSGMASK: 00, <message mask0> \$TTMSGMASK: 01, <message mask1> \$TTMSGMASK: 02, <message mask2> |

| | |
|------------------|---|
| | \$TTMSGMASK: 03, <message mask3> \$TTMSGMASK: 04, <message mask4> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <Message type> | Message type identification (0-4). 0 Keep alive message 1 Position report messages 2 Counter messages 3 Variables messages 4 Geofence messages |
| <Message Mask> | See Message type table for details |
| Notes | The message type defined will be used for all messages send from the device to the server or serial port. Please refer to manual TT8750+ MDT AT Command for more information on other message masks. |

AT\$TTLOGCL Message Log Clear

| | |
|------------------|-----------------------------------|
| AT\$TTLOGCL | Message Log Clear |
| Command Function | This command erases the log data. |
| Query Format | AT\$TTLOGCL =? |
| Response | \$TTLOCL: (0-5) OK |
| Write Format | N/A |
| Response | N/A |
| Read Format | N/A |
| Response | N/A |
| Execution Format | AT\$TTLOGCL=<queue type> |

| | |
|------------------|---|
| <queue type> | 0: All queues 1: UDP only 2: UDP with SMS Backup 3: SMS only 4: TCP only 5: LOG only |
| Response | OK |
| Parameter Values | None |
| Notes | If AT&F is performed; the log buffer will be cleared. |

AT\$TTLOGDMP Dump Messages to Serial Port

| | |
|----------------------|--|
| AT\$TTLOGDMP | Dump Messages to Serial Port |
| Command Function | This command allows the user to dump the contents of the messages to the serial port. This command is non-destructive in that it does not actually remove the messages from the queue |
| Command Format Query | AT\$TTLOGDMP=? |
| Response | \$TTLOGDMP: (0-4),(0-1),(1-83),(0-2) OK |
| Write Format | AT\$TTMSGLOGDMP=<queue>,<format>,<bytes_per_line>,<display_all> |
| Response |// messages output OK |
| Read Format | N/A |
| Response | N/A |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | N/A |
| <queue> | 0 = data that was configured to be sent to a remote server via UDP only 1 = data that was configured to be sent to a remote server via UDP primarily but also use SMS as backup method if GPRS is |

| | |
|------------------|--|
| | <p>not available</p> <p>2 = data that was configured to be sent to a remote server via SMS only</p> <p>3 = data that was configured to be sent to a remote server via TCPAPI only</p> <p>4 = data that was configured to be sent to the log only</p> |
| <format> | <p>0 = ASCII format (if message contains a byte that is not a printable ASCII character, it will be displayed as '?')</p> <p>1 = hex format (Each byte in message is displayed as a two-digit hex character representing the value of the byte with spaces between each byte. Maximum of 16 bytes per line.)</p> |
| <bytes_per_line> | 1-83 (default = 16) number of bytes displayed per line for binary data (each byte is represented as a two-digit hex value followed by a space) |
| <display_all> | <p>0 = display unsent messages only (default)</p> <p>1 = display all messages (unsent and sent) from the indicated queue</p> <p>2= display sent messages from the indicated queue</p> |
| Notes | |

AT\$TTLOGRD Read Log Data

| | |
|------------------|---|
| AT\$TTLOGRD | Read Log Data |
| Command Function | The \$TTLOGRD command is used to query the status of all message logs. Using the queue parameter, this command can be used to resend messages that have already been sent and are still in the log. |
| Query Format | AT\$TTLOGRD=? |
| Response | \$TTLOGRD: (0-4),(0-x),(0-y) OK |
| Write Format | \$TTLOGRD=<queue>,<number of messages>, <starting index> |
| Response | OK |
| Read Format | AT\$TTLOGRD? |

| | |
|-------------------------|--|
| <p>Response</p> | <p>\$TTLOGRD: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0</p> <p>OK</p> <p>AT\$TTLOGRD? command returns 10 values:</p> <p>\$TTLOGRD: udp1,udp2,udpsms1,udpsms2,sms1,sms2,tcp1,tcp2,log1,log2</p> <ul style="list-style-type: none"> • First pair of numbers (udp1, udp2): unsent UDP messages, total number of UDP messages in the UDP message log. • Second pair of numbers (udpsms1, udpsms2): unsent UDP messages with SMS as backup, total number of UDP messages with SMS as backup in the UDP/SMS log. • Third Pair of numbers (sms1, sms2): unsent SMS messages, total number of SMS messages in the SMS log. • Fourth Pair of messages (tcp1, tcp2): unsent messages via TCP, total number of messages in the TCP log. • Fifth pair of messages (log1, log2): Log1 and log2 have the same value and they represent the total number of messages in the log only queue (generated for output event 39). |
| <p>Execution Format</p> | <p>N/A</p> |
| <p>Response</p> | <p>N/A</p> |
| <p>Parameter Values</p> | |
| <p><queue></p> | <p>0 –4 (possible valid values).</p> <p>0 = data that was configured to be sent to a remote server via UDP only.</p> <p>1 = data that was configured to be sent to a remote server via UDP primarily but also use SMS as backup method if UDP is not available.</p> <p>2 = data that was configured to be sent to a remote server via SMS only.</p> <p>3 = data that was configured to be sent to a remote server via TCP.</p> <p>4 = function data that was configured to be sent to the log only.</p> |

| | |
|----------------------|---|
| <number of messages> | Total number of messages one desires to read from the memory. A user can choose to read 1 message in which case x = 1 or read all messages in which case x = 65535. |
| <starting index> | Starting index number of messages that are stored in the memory. |
| Notes | <p><starting index> cannot be greater than total number of stored messages.</p> <p>If AT&F is performed; the message log buffer will be cleared.</p> <p>The message logs work on a FIFO basis. I.e. when a log is full the oldest messages are deleted to make room for new messages.</p> |

AT\$TTSNDMG Send Message

| | |
|------------------|---|
| AT\$TTSNDMG | Send Message |
| Command Function | The \$TTSNDMG command allows to the user send data using all communication methods available in the device. |
| Query Format | AT\$TTSNDMG=? |
| Response | \$TTSNDMG: (0-6),("ASCII DATA"),(0-1) OK |
| Write Format | N/A |
| Response | N/A |
| Read Format | N/A |
| Response | N/A |
| Execution Format | AT\$TTSNDMG=<destination>,<"data">[,<flash>] |
| Response | OK |
| Parameter Values | |
| <destination> | <p>0 – 5 (possible valid values)</p> <p>0 = <"data"> is sent out the serial port</p> <p>1 = <"data"> is sent to all SMS addresses listed in</p> |

| | |
|----------|---|
| | <p>AT\$TTSMSDST command.</p> <p>2 = <"data"> is sent via GPRS to first IP address and port, configured in AT\$TTSRVDST command.</p> <p>3 = <"data"> is sent via GPRS to IP address and Port number listed in the AT\$TTPASPCFG command</p> <p>4 = <"data"> is sent via GPRS to first IP address, configured as server, in AT\$TTSRVDST command and port number for TCP API values</p> <p>5 = <"data"> is sent out the AUX serial port</p> <p>6 = <"data"> is sent via GPRS with ACK to first IP address and port, configured in AT\$TTSRVDST command.</p> |
| <"data"> | Data, a maximum of 248 bytes, ASCII characters, for <destination> 0, 2, 3, 4 and maximum 160 bytes, ASCII characters, for <destination> 1 |
| <flash> | <p>This optional parameter, when enabled, allows a user to store the message in the device's FLASH memory when the device is out of coverage or not registered. Stored messages will be transmitted when the device enters GSM/GPRS coverage.</p> <p>0 – Do not store messages in FLASH memory.</p> <p>1 – Store messages in FLASH memory.</p> |
| Notes | <p>AT\$TTSNDMG command can be sent to the device via SMS, UDP-API, or serial port.</p> <p>AT\$TTSNDMG=4 checks if TCP API is enabled (AT\$TTTCPAPI=1) before attempting to send message. Will report error if not enabled.</p> <p>A carriage return can be added to a \$TTSNDMG message by inserting "\0D" in to the text portion of the message</p> <p>For example: AT\$TTSNDMG=0,"Hello \0D" OK Hello</p> <p>To add a carriage return to a \$TTSTOCMD use the following format:</p> |

| | |
|--|--|
| | <p>AT\$TTSTOCMD=1,AT\$TTMSGSEND=0,"Hello \OD"</p> <p>To send the message with option 6, the device uses the configuration defined with the command AT\$TTACKCFG.</p> |
|--|--|

AT\$TTSMSDST Destination Address for SMS Messages

| | |
|---------------------------|--|
| AT\$TTSMSDST | Destination Address for SMS Messages |
| Command Function | This command allows a user to configure the phone number or email address for the sending of function data. It is also used in limiting the originating address for sending AT commands Over SMS |
| Query Format | AT\$TTSMSDST=? |
| Response | \$TTSMSDST: (1-5),"1234..." ,"123.." OK |
| Write Format | AT\$TTSMSDST=<index>,<dest addr>[,<gateway number>] |
| Response | OK |
| Read Format | AT\$TTSMSDST? |
| Response | \$TTSMSDST: 1,"<dest addr>" ," " \$TTSMSDST: 2,"<dest addr>" ," " \$TTSMSDST: 3,"<dest addr>" ," " \$TTSMSDST: 4,"<dest addr>" ," " \$TTSMSDST: 5,"<dest addr>" ," " OK |
| Execution Format Response | N/A |
| Parameter Values | |
| <index> | 1 – 5 defines the index number for destination address |
| <dest addr> | Phone number or email address, 38 characters or less. |
| Notes | The gateway number is provided by the Network Provider (ex: AT&T, Cingular, etc) and is only used for sending email over |

| | |
|--|--|
| | <p>SMS (*This feature is not implemented yet*). It is not required if you are sending SMS to a phone number.</p> <p>If using this command with an international number (preceded by a "+") it may be required to change the command at+cscs=145.</p> <p>If the TTSMDSST is not defined and use action to send sms, those ones will not be delivery</p> |
|--|--|

AT\$TTWAKEUP Modem to Server Wakeup/Keep Alive

| | |
|------------------|--|
| AT\$TTWAKEUP | Modem to Server Wakeup/Keep Alive |
| Command Function | <p>This command allows the user to configure the modem wakeup/keep alive parameters. These parameters control how the modem initiates contact with its server destinations. Parameters can be selected so that a wakeup message sequence is executed every time the modem receives a new IP, and/or after a requested period has passed since the previous wakeup sequence has completed. A wakeup message sequence consists of sending <max retry> messages to each server destination in sequence (i.e. server 2 is contacted after all retries for server 1 is complete) and is complete when each server destination has received <max retry> messages, or upon receipt of an acknowledge message from a server.</p> |
| Query Format | AT\$TTWAKEUP=? |
| Response | \$WAKEUP: (0-2),(0-10080) OK |
| Write Format | AT\$TTWAKEUP=<wakeup mode>,<retry period> |
| Response | OK |
| Read Format | AT\$TTWAKEUP? |
| Response | \$WAKEUP: <wakeup mode>,<retry period> OK |
| Execution Format | N/A |
| Response | N/A |

| | |
|------------------|--|
| Parameter Values | |
| <wakeup mode> | <p>0 = No wakeup messages sent</p> <p>1 = Send one message upon receipt of new IP and every <retry period> minutes</p> <p>2 = Send acknowledgement message using AT\$TTACKCFG parameters upon receipt of new IP and every <retry period> minutes message</p> |
| <retry period> | The number of minutes for keep alive period. Zero indicates no retries. |
| Notes | <p>When this command is used, it will generate function group 0 in the function table.</p> <p>The <retry period> parameter of this command populates the function timer value when the AT\$TTCNT? command is issued. The AT\$TTCNT value will be in seconds.</p> <p>If AT\$TTFNTDEL=0 is issued or any entry for group 0 is deleted, this command MUST be re-entered for proper functionality. If a read command is issued, it will not reflect the true state of the AT\$TTWAKEUP setting.</p> <p>Wakeup messages are sent to the IPs and port specified in AT\$TTSRVDST.</p> |

AT%BYCNT Query (or Reset) the Byte Counters (GPRS Only)*

| | |
|------------------|--|
| AT%BYCNT | Query (or Reset) the Byte Counters. (GPRS Only) |
| Command Function | Returns (or resets) the byte counts of every current connection. |
| Query Format | AT%BYCNT=? |
| Response | %BYCNT: (0) OK |
| Write Format | AT%BYCNT=<rst> |
| Response | OK |
| Read Format | AT%BYCNT? |

| | |
|------------------|---|
| Response | %BYCNT: <nsapi1>,<upo>,<dno>,<upp>,<dn><CR><LF> %BYCNT: <nsapi2>,<upo>,<dno>,<upp>,<dn><CR><LF> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <rst> | resets the counters if rst = 0 |
| <nsapi> | connection id |
| <upo> | uplink octets count. |
| <dno> | downlink octets count. |
| <upp> | uplink packets count. |
| <dn> | downlink packets count. |
| Notes | If multiple contexts are activated / deactivated, set/query will result in OK followed by byte counts (set command resets the counters to 0) for each connection. * In development and subject to change |

API Commands

TCP API Commands

AT\$TTTCPAPI TCP API Control

| | |
|------------------|--|
| AT\$TTTCPAPI | TCP API Control |
| Command Function | This command allows the user to initiate and terminate and query the status of the TCP API connection for the first server |
| Query Format | AT\$TTTCPAPI=? |
| Response | \$TCPAPI: (0-1) OK |
| Write Format | AT\$TTTCPAPI=<Status> |
| Response | OK |
| Read Format | AT\$TTTCPAPI? |
| Response | \$TTTCPAPI: <status>(M-<Mgr Task>,<Destination Index>) OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <Status> | TCP API connection status 0=Disabled 1=Enabled |
| <Mgr Task> | TCP API Manager Task 0 = None 1 = Init 2 = Idle 3 = Connecting 4 = Connected |

| | |
|---------------------|----------------------------|
| | 5 = Disconnecting |
| <Destination Index> | Destination Index (1 – 10) |
| Notes | |

AT\$TTTCPCFG TCP API Configuration

| | |
|---------------------|---|
| AT\$TTTCPCFG | TCP API Configuration |
| Command Function | Configure the parameters used for TCP connection |
| Query Format | AT\$TTTCPCFG=? |
| Response | \$TCPCFG: (10-86400), (120-65535), (1024-65535),(1024-65535) OK |
| Write Format | AT\$TTTCPCFG=<Timeout>, <Retry Timeout>,<Start Port Number>,<End Port Number> |
| | OK |
| Read Format | AT\$TTTCPCFG? |
| Response | \$TTTCPCFG: <Idle Timeout>, <Retry Timeout>,<Start Port Number>,<End Port Number> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <Idle Timeout> | TCP API idle timeout value. Specifies the number of seconds without data traffic, in either direction, before closing the connection. |
| <Retry Timeout> | TCP API retry timeout value. Specifies the number of seconds without receiving a TCP level ACK that will cause the connection to be closed. |
| <Start Port Number> | TCP API starting port number. Specifies the TCP API source port range used when making a TCPAPI connection. |
| <End Port Number> | TCP API ending port number |
| Notes | After closing the connection, the device will attempt to reconnect using the Server list. |

| | |
|--|--|
| | <p>Currently, the number of retries is 10 and the amount of time varies based on calculated round trip time. The minimum time allowed is 120 seconds.</p> <p>Attempts to set the retry timeout to a value less than 120 or more than 65535 will result in an error.</p> <p>Each connection attempt uses the next port number in sequence until the end port is passed. When this happens the port is set to the start port number.</p> <p>This current port number in use is retained over a power cycle.</p> <p>If only the start port number is provided, the end port number will be start port number + 49 (range of 50).</p> <p>If not start port number is provided the modem will use the default start port number of 1024. The modem will incrementing the port number with each connection attempt until it reaches the default end port number of 65535. Once the modem reaches the end port number it will begin incrementing the ports again using the start port number of 1024.</p> |
|--|--|

AT\$TTTCPRST TCP API Restart

| | |
|------------------|--|
| AT\$TTTCPRST | TCP API Restart |
| Command Function | If a connection exists, it is dropped and a new connection is attempted starting at the beginning of the Destination list. |
| Query Format | AT\$TTTCPRST=? |
| Response | OK |
| Write Format | N/A |
| Read Format | N/A |
| Response | N/A |
| Execution Format | AT\$TTTCPRST |
| Response | OK |
| Parameter Values | N/A |
| Notes | N/A |

AT\$TTTCCNT TCP API Statistics*

| | |
|----------------------|---|
| AT\$TTTCCNT | TCP API Statistics |
| Command Function | Displays bytes transmitted and received since last reset or last AT\$TTTCCNT=0 command. |
| Query Format | AT\$TTTCCNT=? |
| Response | \$TCCNT: (0) OK |
| Write Format | AT\$TTTCCNT =0 |
| Response | OK |
| Read Format | AT\$TTTCCNT? |
| Response | \$TTTCCNT: <Rx Bytes>,<Tx Bytes>,<Mode Change>,<GPRS Deactivate>,<Restarts>,<Connection Timeout>,<Idle Timeout>,<Socket Errors> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <Rx Bytes> | TCP API bytes received |
| <Tx Bytes> | TCP API bytes transmitted |
| <Mode Changes> | Mode change (AT\$TTTCCAPI=0) |
| <GPRS Deactivate> | GPRS deactivate |
| <Restarts> | TCP API restarts (AT\$TTTCCPRST) |
| <Connection Timeout> | TCP API connection timeout |
| <Idle Timeout> | TCP API idle timeout |
| <Socket Errors> | TCP API socket errors |
| Notes | AT\$TTTCCNT=0 will clear all TCP API statistics. * In development and subject to change |

UDP API Commands

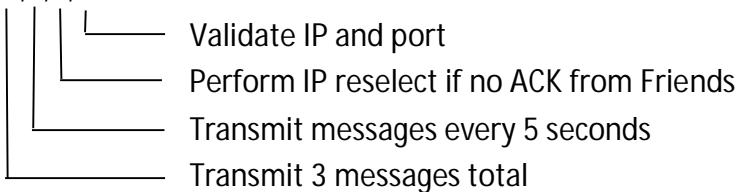
AT\$TTACKCFG Acknowledgment Message Period & Retry Number

| | |
|------------------|---|
| AT\$TTACKCFG | Acknowledgment Message Period & Retry Number |
| Command Function | This command allows the user to configure the modem message acknowledge behavior. If server acknowledgement is selected for a message, the message will be re-sent every <retry period> number of seconds until the acknowledge message sequence is complete, or until an acknowledge message is received from a server. An acknowledge message sequence consists of sending <max retry> messages to each server destination in sequence (i.e. server 2 is contacted after all retries for server 1 is complete) and is complete when each server destination has received <max retry> messages, or upon receipt of an acknowledge message from a server. |
| Query Format | AT\$TTACKCFG=? |
| Response | \$TTACKCFG: (0-255),(0-3600),(0,1),(0,1) OK |
| Write Format | AT\$TTACKCFG=<max retry>,<retry period>,<IP reselect>,<port need> |
| Response | OK |
| Read Format | AT\$TTACKCFG? |
| Response | \$TTACKCFG: <max retry>,<retry period>,<IP reselect>,<port need> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <max retry> | The maximum number of times an acknowledge message is re-sent to a single destination server. After all retries to the destination server are exhausted, the modem will move on to the next destination server if one exists. If there are no more destination servers available, the modem will start PDP activation recovery if the recovery option is selected; |

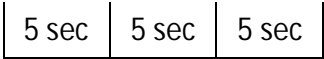
| | |
|----------------|--|
| | <p>otherwise, the message will be discarded.</p> <p>In the case of the default acknowledge wakeup message: The maximum number of wakeup messages the modem will send to each server destination upon receipt of a new IP, or upon expiration of each keep-alive period. Zero indicates no wakeup message should be sent.</p> |
| <retry period> | The number of seconds between successive messages retries. Zero indicates no retries. |
| <IP reselect> | <p>0 = IP reselection is OFF. The message will be removed from the queue.</p> <p>1 = If an acknowledge message has not been received after all destination servers and retries for the message are exhausted, assume a problem with round-trip communication and initiate IP re-selection after it has reestablish the gprs connection, the device restarts the process.</p> |
| <port need> | <p>A numeric to indicate whether the filter sever with port.</p> <p>0: Do not filter a UDP server with port number.</p> <p>1: Filter a UDP server with both IP address and port number. The server must send commands from the same port number that it received the reports from the device.</p> |
| Notes | N/A |

Examples

AT\$TTACKCFG =3,5,1,1



AT\$TTWAKEUP time between AT\$TTACKCFG sequence



AT\$TTACKCFG Sending 3 messages, 5 seconds apart

PAD Commands

AT\$AUXURTCFG Auxiliary Serial Port Configuration

| | |
|------------------|---|
| AT\$AUXURTCFG | Configure the properties of the auxiliary serial port |
| Command Function | This command allows the user to query/set the properties of the auxiliary serial port |
| Query Format_ | AT\$AUXURTCFG =? |
| Response | \$AUXURTCFG: (1200,2400,4800,9600,14400,19200,28800,38400,57600, 115200), (7-8), (0-3), (1-3) OK |
| Write Format | AT\$AUXURTCFG =<baud_rate>, <dat_bit>, <parity>, <stop_bit> |
| Response_ | OK |
| Read Format | AT\$AUXURTCFG? |
| Response | \$AUXURTCFG: <baud_rate>, <dat_bit>, <parity>, <stop_bit> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <baud_rate> | The baud rate for the auxiliary serial port |
| <data_bit> | Data bit of the auxiliary serial port |
| <parity> | Parity of the auxiliary serial port 0 = Odd 1 = Even 2 = Mark (1) 3 = Space (0) |
| <stop_bit > | Stop bit of the auxiliary serial port 1 = one stop bit |

| | |
|-------|--------------------------------------|
| | 2 = 1.5 stop bits 3 = 2 stop bits |
| Notes | |

AT\$TTPASPCFG PAD State Configuration

| | |
|----------------------|--|
| AT\$TTPASPCFG | PAD State |
| Command Function | This command allows the user to query/set PAD modem configuration |
| Query Format_ | AT\$TTPASPCFG=? |
| Response | \$TTPASPCFG: (1-1), (0, 10-3600), (10-86400), (3-1472), (0-FF), "(0-255),(0-255),(0-255),(0-255)", (0-65535), (0-FF), (0-FF), (0- 65535) OK |
| Write Format | AT\$TTPASPCFG =<state>, <connection timeout>, <idle timeout>, <block size>, <backspace character>,"<PAD destination IP> or <PAD destination DNS name>",<PAD destination port>,<escape character>,<forward character>,<PAD timeout> |
| Response_ | OK |
| Read Format | AT\$TTPASPCFG? |
| Response | \$TTPASPCFG: <state>, <connection timeout>, <idle timeout>, <block size>, <backspace character>,"<PAD destination IP> <PAD destination DNS name>",<PAD destination port>,<escape character>,<forward character>,<PAD timeout> |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <state> | This command determines the active or passive state of the TCP PAD connection. 1 = PAD active/client mode |
| <connection timeout> | PAD Connection Timeout 0 = Infinite timeout value |

| | |
|----------------------------|--|
| | 10-3600 = timeout value in seconds |
| <idle timeout> | Period of time, in seconds, a TCP/UDP session connection will remain active without the remote connection sending any data. 10-86400 = timeout value in seconds |
| <block size> | PAD data will be created at the requested PAD block size (number of bytes) unless an enabled forward character or PAD timeout forces the data to be sent out at a smaller block size. Block size does NOT include the IP or TCP/UDP header size. Block size values over 1460 are truncated to 1460 for TCP PAD. Values over 1460 are only applicable to UDP PAD. |
| <backspace character> | Hex representation of user selected backspace character. Normal backspace character is 08. |
| <PAD destination IP> | Destination IP for PAD data. PAD data is sent to and received from this IP. A destination IP address of 0 will allow PAD access from any IP destination, and will cause all locally generated PAD data to be sent to the IP address associated with the last remotely received PAD data. |
| <PAD destination DNS name> | Destination DNS name for PAD data. |
| <PAD destination port> | Destination port for PAD data. PAD data is sent to and received from this port. A destination port of 0 will allow PAD access from any port, and will cause all locally generated PAD data to be sent to the port associated with the last remotely received PAD data. |
| <escape character> | Hex representation of user selected escape character. Default escape character is 00 |
| <forward character> | Hex representation of user selected forward character. Default forward character is 0D (Carriage return). |
| <PAD timeout> | The number of tenths of seconds to wait for the receipt of more PAD data before forwarding the currently accumulated PAD buffer to the PAD destination. A value of zero disables the PAD timeout feature. If the PAD timeout feature is disabled, no data will be forwarded to the destination until either an enabled forward character is received, or the selected PAD buffer size is reached. (50 = 5 seconds) |
| Notes | The PAD will be in client mode and will initiate a connection |

| | |
|--|--|
| | <p>based on the ATDT command</p> <p>Connection timeout equal 0 will indicate infinite connection wait time.</p> <p>If no communication is received from the remote connection in the specified idle time, the modem will gracefully attempt to close the connection.</p> <p>You will use either the PAD Destination IP Address, or the PAD Destination DNS Name, but not both.</p> |
|--|--|

AT\$TTPASPCMF PAD Command Features

| | |
|----------------------|--|
| AT\$TTPASPCMF | PAD Command Features |
| Command Function | This command allows the user to set/query PAD configuration options. |
| Query Format | AT\$TTPASPCMF=? |
| Response | \$TTPASPCMF: (0-FFFF) OK |
| Write Format | AT\$TTPASPCMF=<pad feature select> |
| Response | OK |
| Read Format | AT\$TTPASPCMF? |
| Response | \$TTPASPCMF: "<pad feature select>" |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <pad feature select> | <p>AND selected HEX options into a single 16 bit word.</p> <p>0001 = Enable forwarding on special char</p> <p>0002 = Forward special char with data</p> <p>0008 = Enable backspace</p> <p>0400 = Enable escape character to treat next character as data</p> |
| Notes | A +++ is an escape sequence to exit PAD mode. Disabling of the escape sequence is not supported, however the escape is only applicable when there is a 1 second guard time before and |

| | |
|--|---|
| | after the +++. If the guard period is not met before and after the escape sequence, it will be forwarded as data. |
|--|---|

FOTA Commands

AT\$TTSFOTA Upgrade Firmware OTA

| | |
|---------------------|--|
| AT\$TTSFOTA | Upgrade Firmware over the air |
| Command Function | This function is used to update the firmware over the air. |
| Query Format | AT\$TTSFOTA=? |
| Response | \$TTSFOTA: (0-20),(1-60),(0),("URL") OK |
| Write Format | N/A |
| Response | N/A |
| Read Format | AT\$TTSFOTA=? |
| Response | OK |
| Execution Format | AT\$TTSFOTA=<max retry>,<download timeout>,<download protocol>,<URL> |
| Response | OK |
| Parameter Values | |
| <max retry> | Maximum number of retry |
| <download timeout> | Maximum time (in seconds) used by the device to download the file |
| <download protocol> | Protocol used to download the file. This time is supported only http (0) |
| <URL> | Server address from where the device will get the new firmware file |
| Notes | |

FTP Commands

AT\$TTFSA Generic Flash File System Access*

| | |
|----------------------|---|
| AT\$TTFSA | Generic Flash File System Access |
| Command Function | This command provides generic file read and write functionality to the embedded FSA in the Skypatrol modem. |
| Command Format Query | AT\$TTFSA=? |
| Response | \$TTFSA: (0-5,86),"filename",(0-524287),(0-150),("ASCII-HEX") |
| Write Format | AT\$TTFSA=<op>,<filename>,<seek>,<length>,<data> |
| Response | <p>Response varies by <op></p> <p>Write: OK</p> <p>Read: \$TTFSA: <data> OK</p> <p>Delete: OK</p> <p>Directory: \$TTFSA [<filename>] [<filename>] [...] OK</p> <p>Directory and size: \$TTFSA: [<file size> <filename>]</p> |

| | |
|------------------|--|
| | <p>[<file size> <filename>] [...] Free and used space: \$TTFSA: <free_bytes> Free Bytes, <used_bytes> Used Bytes</p> <p>Delete All: OK</p> |
| Read Format | N/A |
| Response | N/A |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <op> | <p>TTFSA Operation</p> <p>0 = write 1 = read 2 = delete 3 = directory 4 = directory with file size 5 = FSS free space and used space 86 = delete all files</p> |
| <filename> | <p>The filename in double quotes (i.e. "test.txt") Maximum length is 16 characters. Filename parameter is required for the write (0), read (1), and delete (2) operations.</p> |
| <seek> | <p>The position in the file to read or write given as a byte offset. Seek parameter is required for the write (0) and read (1), operations. Allowed values range from 0 to 524287.</p> |
| <length> | <p>The number of bytes to read or write. Length parameter is only required for read and write operations. Allowed values range from 0 to 150.</p> |
| <data> | <p>Only applicable to write operations as an input parameter. Also defines the output data format for the read operation.</p> |

| | |
|----------|---|
| | Limited to 150 bytes (300 characters). Data is encoded in ASCII hex (i.e. 0x12, 0x34, 0xab would be entered as "1234ab") |
| Notes | * In development and subject to change |
| Examples | <pre> AT\$TTFSA=0,"test",0,8,"0102030405060708" OK AT\$TTFSA=1,"test",0,8 \$TTFSA: 0102030405060708 OK AT\$TTFSA=3 \$TTFSA: test OK AT\$TTFSA=4 \$TTFSA: 8 test OK AT\$TTFSA=5 \$FFS: 675308 Free Bytes, 76324 Used Bytes OK AT\$TTFSA=2,"test" OK AT\$TTFSA=86 OK </pre> |

AT\$TTFTPCN Cancels Current Data Operation with FTP Server*

| | |
|----------------------|--|
| AT\$TTFTPCN | Cancels Current Data Operation with FTP Server |
| Command Function | Interrupts the current data operation with the FTP server. |
| Command Format Query | AT\$TTFTPCN=? |
| Response | OK |
| Write Format | N/A |
| Response | N/A |
| Read Format | AT\$TTFTPCN? |
| Response | ERROR |
| Execution Format | AT\$TTFTPCN |
| Response | OK |
| Parameter Values | N/A |
| Notes | <p>If a data operation is not in operation, no action is performed. If a connection to an FTP server is not active, FTP INVALID CLIENT is returned.</p> <p>* In development and subject to change</p> |

AT\$TTFTPDEF FTP Definition*

| | |
|----------------------|---|
| AT\$TTFTPDEF | Define the parameters used for the FTP connection |
| Command Function | Changes the current FTP block size |
| Command Format Query | AT\$TTFTPDEF=? |
| Response | <p>\$TTFTPDEF: (128-1500), "host", (0-65535), "username", "password"</p> <p>OK</p> |
| Write Format | AT\$TTFTPDEF=<blocksize>, <host>,<portnum>,<username>,<password> |
| Response | OK |

| | |
|------------------|---|
| Read Format | AT\$TTFTPDEF? |
| Response | TTFTPDEF : <blocksize>, <host>,<portnum>,<username>,<password> |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <blocksize> | FTP block size (128-1500) |
| <host> | Fully qualified domain name or dotted-decimal notation IP address of FTP server, character string up to 63 characters |
| <portnum> | FTP server port number, 0 – 65535 |
| <username> | FTP server user name, character string up to 31 characters |
| <password> | FTP server password, character string up to 31 characters |
| Notes | <p>The default blocksize is 1500. AT\$TTFTPDEF should be set prior to the AT\$TTFTPOP for the blocksize to have effect.</p> <p>All parameters must be entered initially before opening connection to FTP server. Configuration information will be saved to flash when flash write command is executed and will be re-used on subsequent FTP connections.</p> <p>* In development and subject to change</p> |

AT\$TTFTPCD Change current working directory*

| | |
|----------------------|--|
| AT\$TTFTPCD | Change current working directory |
| Command Function | Changes the current working directory on the FTP server. |
| Command Format Query | AT\$TTFTPCD=? |
| Response | \$TTFTPCD: "directory" OK |
| Write Format | AT\$TTFTPCD=<directory> |
| Response | OK |
| Read Format | AT\$TTFTPCD? |
| Response | ERROR |

| | |
|------------------|---|
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <directory> | Directory name on remote FTP server |
| Notes | AT\$TTFTPCD can only be entered after the connection to the FTP server has been established. To navigate up a directory, use the ".." directory notation. * In development and subject to change |

AT\$TTFTPCL Closes the Connection to the FTP Server*

| | |
|----------------------|---|
| AT\$TTFTPCL | Closes the Connection to FTP Server |
| Command Function | Closes the current connection with the remote FTP server. |
| Command Format Query | AT\$TTFTPCL=? |
| Response | OK |
| Write Format | N/A |
| Response | N/A |
| Read Format | AT\$TTFTPCL? |
| Response | ERROR |
| Execution Format | AT\$TTFTPCL |
| Response | OK |
| Parameter Values | N/A |
| Notes | This command is only available when a connection to the FTP server is active. If a connection to an FTP server is not active, FTP INVALID CLIENT is returned. * In development and subject to change |

AT\$TTFTPDIR Directory Listing*

| | |
|--------------|-------------------|
| AT\$TTFTPDIR | Directory Listing |
|--------------|-------------------|

| | |
|----------------------|---|
| Command Function | Prints a directory listing of the current working directory on the FTP server |
| Command Format Query | AT\$TTFTPDIR=? |
| Response | OK |
| Write Format | N/A |
| Response | N/A |
| Read Format | AT\$TTFTPDIR? |
| Response | OK \$FTPDIR: <current directory information> |
| Execution Format | AT\$TTFTPDIR |
| Response | OK <directory listing information> |
| Parameter Values | N/A |
| Notes | AT\$TTFTPDIR can only be entered after the connection to the FTP server has been established. Directory listing information may be delayed due to the latency of the data connection to the FTP server. * In development and subject to change |

AT\$TTFTPRT Retrieve a Remote File*

| | |
|----------------------|---|
| AT\$TTFTPRT | Retrieve a Remote File |
| Command Function | Retrieves a remote file from the FTP server for either output on the serial port or storage to the local flash file system. |
| Command Format Query | AT\$TTFTPRT=? |
| Response | \$TTFTPRT: "remotefile", "localfile" OK |
| Write Format | AT\$TTFTPRT=<remotefile>,<localfile> |
| Response | OK |
| Read Format | AT\$TTFTPRT? |
| Response | \$TTFTPRT: <remotefile>,(SERIAL, |

| | |
|------------------|---|
| | <localfile>),<output>/<available>,<eof> |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <remotefile> | File name on the remote FTP server. |
| <localfile> | File name to use for storage of the retrieved file in the local flash file system. IF the file name is not specified the remote file will be copied to serial port |
| <output> | Number of bytes which have been transferred out the serial port |
| <available> | Number of bytes of file which are available for output to the serial port |
| <eof> | 0 = file currently being transferred 1 = complete file has been transferred to FTP client |
| Notes | If local file is not specified, the remote file will be maintained in memory buffers until a subsequent FTP read request is performed to output a block of bytes onto the serial port. As the file is transmitted out the serial port, the FTP client will refill the memory buffers with additional file data from the FTP server. * In development and subject to change |

AT\$TTFTPOP Opens FTP Connection*

| | |
|----------------------|--|
| AT\$TTFTPOP | Opens FTP Connection |
| Command Function | Opens FTP control connection to FTP server |
| Command Format Query | AT\$TTFTPOP=? |
| Response | \$TTFTPOP: ("ACTIVE","PASSIVE") OK |
| Write Format | N/A |
| Response | N/A |
| Read Format | AT\$TTFTPOP? |

| | |
|------------------|--|
| Response | OK \$TTFTPOP: <status> |
| Execution Format | AT\$TTFTPOP="<mode>" |
| Response | OK <status> |
| Parameter Values | |
| <mode> | ACTIVE or PASSIVE |
| | Mode is an optional parameter. If not specified, the FTP connection will be in the ACTIVE mode. |
| <status> | FTP CONNECTION OPEN FTP CONNECTION CLOSED FTP STACK ERROR Status will display the state of the control connection with the remote FTP server. |
| Notes | AT\$TTFTPOP must be entered after FTP configuration information has been entered. If any FTP configuration information has been saved to flash, it will be used to open the connection to the FTP server. After the FTP open completes successfully, FTP CONNECTION OPEN will be printed, otherwise an error string will be printed. * In development and subject to change |

AT\$TTFTPSTR Store a Local File on the FTP Server*

| | |
|----------------------|--|
| AT\$TTFTPSTR | Store a Local File On the FTP Server |
| Command Function | Stores a local file from the Flash File System onto the remote FTP server. |
| Command Format Query | AT\$TTFTPSTR=? |
| Response | \$TTFTPSTR:"remotefile", "localfile" OK |
| Write Format | AT\$TTFTPSTR=<remotefile>,<localfile> |
| Response | OK |

| | |
|------------------|---|
| Read Format | AT\$TTFTPSTR? |
| Response | \$TTFTPSTR:<remotefile>,<localfile>,<totalWrote>/<totalSize>,<eof> |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <remotefile> | File name on the remote FTP server. |
| <localfile> | File name on the local Flash File System. |
| <totalWrote> | Number of bytes of file which have been written to FTP Server |
| <totalSize> | Size of file to be transferred to FTP Server (in bytes). |
| <eof> | 0 = file is currently being transferred 1 = complete file has been transferred to FTP Server |
| Notes | * In development and subject to change |

AT\$TTFTPST Restart the Last FTP Retrieve Operation*

| | |
|----------------------|---|
| AT\$TTFTPST | Restart the Last FTP Retrieve Operation |
| Command Function | Manual restart of an FTP Retrieve files transfer from the last completed point in the file. |
| Command Format Query | N/A |
| Response | N/A |
| Write Format | AT\$TTFTPST=<offset> |
| Response | OK |
| Read Format | N/A |
| Response | N/A |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <offset> | Number of bytes into the file to resume file transfer. (Optional) |

| | |
|-------|---|
| Notes | <p>AT\$TTFTPRST will attempt to resume the last incomplete file transfer from the last point in the file successfully read. Offset is not required unless the automatically computed offset is to be overridden. The AT\$TTFTPRST command requires an RFC3659 compliant FTP server which implements the FTP REST command. AT\$TTFTPRST is not designed to work over a system restart.</p> <p>The user's current path must contain the file prior to issuing AT\$TTFTPRST or else the AT\$TTFTPRST command will fail.</p> <p>The only time it is necessary to send the \$FTPRST command is if a "FTP TIMEOUT" URC is received during a FTP transfer. The "FTP TIMEOUT" is typically received due loss of GPRS network connection for more than the FTP idle timeout, 60 seconds.</p> <p>* In development and subject to change</p> |
|-------|---|

Real-Time Clock Commands

AT\$TTRTCAL Real Time Clock Alarm

| | |
|------------------|--|
| AT\$TTRTCAL | Real Time Clock Alarm |
| Command Function | This command handles the setting and querying of the RTC alarm registers. When the alarm event feature has been enabled the \$TTFNT engine, it will be invoked upon the alarm time. |
| Query Format | AT\$TTRTCAL=? |
| Response | \$TTRTCAL: (1)(0-99),(1-12),(1-31),(0-23),(0-59),(0-59),(0-527040) OK |
| Write Format | AT\$TTRTCAL=<rtc_index><rtc_year>,<rtc_month>,<rtc_day>,<rtc_hour>,<rtc_min>,<rtc_sec>,<rtc_alarmTimeinMinutes> |
| Response | OK |
| Read Format | AT\$TTRTCAL? |
| Response | \$TTRTCAL: <rtc_index><rtc_enabled>,<rtc_year>,<rtc_month>,<rtc_day>,<rtc_hour>,<rtc_min>,<rtc_sec>,<rtc_alarmTimeinMinutes>" OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | Parameters are positional dependent, any parameter may be omitted with the use of the comma (',') as a place holder on command line. If a parameter is omitted then the current value in the hardware is used. |
| <rtc_index> | This is the position of the alarm (only one alarm supported). |
| <rtc_enabled> | Indicates if alarm is enabled or not. 1= Enabled 0= Disabled |
| <rtc_year> | The year on which the alarm is being set to trigger. The RTC |

| | |
|--------------------------|---|
| | supports years 2000-2099. The data is entered as a two digit value 0..99. |
| <rtc_month> | The month on which the alarm is being set to trigger. Values range from 1..12. |
| <rtc_day> | The day on which the alarm is being set to trigger. Values range from 1..31. |
| <rtc_hour> | The hour on which the alarm is being set to trigger. Values range from 0..24 for 24-Hour mode settings. NOTE: only 24-Hour mode currently supported. |
| <rtc_min> | The minute on which the alarm is being set to trigger. Values range from 0..59. |
| <rtc_sec> | The second on which the alarm is being set to trigger. Values range from 0..59. |
| <rtc_alarmTimeinMinutes> | Periodic Alarm time in minutes. RTC Alarm will be reset at a period specified by this parameter. |
| Notes | This command is used to set the Alarm time for the RTC. Currently all time is based on 24-Hour time format. No checks are made for alarm time not being later than current time. AT\$TTRTCAL will not trigger if the alarm time occurs while the device is resetting. |
| Examples | Following sets and alarm for 2012, October, 15th at 19:00 Hours AT\$TTRTCAL=12,10,15,19,0,0 OK |

AT\$TTRTCTI Real Time Clock

| | |
|------------------|--|
| AT\$TTRTCTI | Real Time Clock |
| Command Function | This command handles the setting and querying of the RTC time. |
| Query Format | AT\$TTRTCTI=? |
| Response | \$RTCTI: (0-6),(0-99),(1-12),(1-31),(0-23),(0-59),(0-59) OK |
| Write Format | N/A |

| | |
|------------------|--|
| Response | OK |
| Read Format | AT\$TTRTCTI? |
| Response | \$RTCTI: <rtc_wkday>, <rtc_year>, <rtc_month>, <rtc_day>, <rtc_hour>, <rtc_min>, <rtc_sec>" OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | Parameters are positional dependent, any parameter may be omitted with the use of the comma (',') as a place holder on command line. If a parameter is omitted then the current value in the hardware is used. |
| <rtc_wkday> | Current week day matching time day being set The week day values range from 0-6, where; 0 = Sunday 1 = Monday 2 = Tuesday 3 = Wednesday 4 = Thursday 5 = Friday 6 = Saturday |
| <rtc_year> | The year to which the time is being set. The RTC supports years 2000-2099. The data is entered as a two digit value 0-99. |
| <rtc_month> | The month to which the time is being set. Values range from 1-12. |
| <rtc_day> | The day to which the time is being set. Values range from 1-31. |
| <rtc_hour> | The hour to which the time is being set. Values range from 0-24 for 24-Hour mode settings. NOTE: only 24-Hour mode currently supported. |
| <rtc_min> | The minute to which the time is being set. Values range from 0-59. |
| <rtc_sec> | The second to which the time is being set. Values range from 0-59. |
| Notes | This command is used to set the time for the RTC. Currently all time is based on 24-Hour time format. |

| | |
|----------|--|
| Examples | AT\$TTRTCTI? \$RTCTI: 01, 12, 03, 27, 15, 52, 0 OK AT\$TTRTCTI=? \$RTCTIME: (0..6), (0..99), (1..12), (1..31), (0..23), (0..59), (0..59) OK |
|----------|--|

AT\$TTDOW Day of the Week

| | |
|------------------|--|
| AT\$TTRTCTI | Date of the week |
| Command Function | This command define the day of the week alerts |
| Query Format | AT\$TTDOW=? |
| Response | \$TTDOW: (1-127) OK |
| Write Format | N/A |
| Response | OK |
| Read Format | AT\$TTDOW? |
| Response | \$TTDOW: <dow> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <dows> | <p>This field is the sum of the days of the weeks desired. A user has the following message options to select from. Maximum value for <dows> in this case would be 127.</p> <p>The week day values range from 0-6, where;</p> <ul style="list-style-type: none"> 1 = Sunday 2 = Monday 4 = Tuesday 8 = Wednesday 16 = Thursday 32 = Friday |

| | |
|----------|---------------|
| | 64 = Saturday |
| Notes | N/A |
| Examples | |

Network Commands

AT\$TTARG Auto Registration

| | |
|----------------------|--|
| AT\$TTARG | Auto Registration |
| Command Function | This command sets the auto registration state of the modem |
| Command Format Query | AT\$TTARG=? |
| Response | \$TTARG: (0-3), (0-4) OK |
| Write Format | AT\$TTARG=<state>,<backoff table> |
| Response | OK |
| Read Format | AT\$TTARG? |
| Response | If TTARG is 0-2 \$TTARG: <state> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <state> | 0 = Autoreg off 1 = Autoreg on 2 = Auto GPRS Activation on Power up. (for \$PASP=1 and 2, device will perform GPRS activation and go into PAD data mode. For \$TTPASP=0, device will perform GPRS activation, but remain in AT command mode) 3 = Auto GPRS Activation on Power up. In TCP it will reestablish the TCP session automatically |
| <backoff table> | |
| Notes | This command sets GSM registration state. When set to 1, upon power on, the modem will automatically register on the GSM network. To set the modem to automatically attach to the GPRS network on power on, see AT%CGATT command. |

| | |
|--|--|
| | <p>AT+CGDCONT must be entered and saved before device is placed in ARG=2 or ARG=3.</p> <p>If PIN is enabled, the device will not complete the auto registration process until after the PIN has been entered (AT+CPIN).</p> <p>Do not use AT\$TTARG=2 or AT\$TTARG=3 with autobauding of the serial port and PAD functions. The serial port will not respond to at commands if the modem establishes a connect state before the baud rate has been determined for the serial port.</p> |
|--|--|

AT\$TTTOLST Tower List*

| | |
|----------------------|--|
| AT\$TTTOLST | Tower List |
| Command Function | This command will display the top three channels of each network found. |
| Command Format Query | AT\$TTTOLST=? |
| Response | OK |
| Write Format | N/A |
| Response | N/A |
| Read Format | N/A |
| Response | N/A |
| Execution Format | AT\$TTTOLST |
| Response | <pre> \$TOLST: MCC: 310 MNC: 260 LAC CellId ARFCN RSSI 27b0 3414 747 24 27b0 3413 740 18 MCC: 310 MNC: 410 LAC CellId ARFCN RSSI Cb23 7f9 232 19 Cb23 41c 160 18 Cb23 160 248 18 </pre> |

| | |
|------------------|--|
| | OK |
| Parameter Values | N/A |
| Notes | <p>This command may take up to 5 minutes to complete. If the modem is registered, it will deregister and remain deregistered until the AT\$TTTOLST is complete. Also if modem has an active PDP context, the PDP context will be deactivated and remain deactivated until the AT\$TTTOLST is complete. After the AT\$TTTOLST is complete, the modem will re-register and re-activate the PDP context depending on user settings.</p> <p>* In development and subject to change</p> |

AT\$TTNETWD Network Watchdog

| | |
|-------------------|--|
| AT\$TTNETWD | Network Watchdog |
| Command Function | This command allows the modem to take aggressive network recovery action based upon the results of continuous network monitoring. |
| Query Format | AT\$TTNETWD=? |
| Response | \$TTNETWD: (0,5-1440),(0-10),(0-255),(0-3) OK |
| Write Format | AT\$TTNETWD= <net_unavail_min>,<reset_cnt>,<ping check>,<rst timers> |
| Response | OK |
| Read Format | AT\$TTNETWD? |
| Response | AT\$TTNETWD: "<net_unavail_min>,<reset_cnt>,<ping check>,<rst timers>" OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <net_unavail_min> | Number of minutes the network must remain unavailable before current GPRS Activation is released, and a new GPRS Activation is attempted. Network availability is determined by monitoring GPRS attach status (AT%CGREG) and valid Network |

| | |
|--------------|---|
| | IP (AT\$TTNETIP). A value of zero means the GPRS Activation will never be released via AT\$TTNETWD. |
| <reset_cnt> | Number of GPRS Activations attempted before all volatile network knowledge is erased and the modem performs a soft reset. A value of 1 indicates the modem will perform a graceful detach from the network and then a soft reset of the device. For values greater than 1, the modem will attempt a GPRS deactivation / activation sequence every <net_unavail_min> until the number of attempts equals <reset_cnt>. The modem then will perform a graceful detach from the network and then a soft reset. A value of zero indicates that a modem reset will never occur via AT\$TTNETWD. |
| <ping check> | Number of minutes between modem-initiated ping checks. If no network data has been received within <ping check> minutes, the modem will initiate pings (up to 4 ICMP messages are generated) to the 1st server on the \$TTSRVDST list. If no ping response is received to any of the 4 ICMP messages, the modem will initiate pings to the next server in the list. If no ping response is returned from any of the \$TTSRVDST servers, a new IP is obtained via a modem-initiated GPRS de-activation / activation sequence. A value of zero indicates that the modem will never initiate a ping check. |
| <rst timers> | 0 = Reset network monitoring timers upon any activity on the serial port AND use legacy IP reselect for <ping check> recovery 1 = Do not reset the network monitoring timers if there is activity on the serial ports 2 = Use modem reset for <ping check> recovery 3 = Do not reset the network monitoring timers if there is activity on the serial ports AND use modem reset for <ping check> recovery |
| Notes | This command is intended for extreme activation conditions, such as repeatedly moving in and out of coverage areas, or for modems that are required to be attached to the network continuously. When the net_unavail_min is not zero and \$TTARG=2 or \$TTARG=3 is in effect. |

AT\$TTGSMJDC Jamming Detection

| | |
|--------------------|---|
| AT\$TTGSMJDC | Jamming Detection |
| Command Function | This command allows the user to set the GSM Jamming detection parameters. |
| Query Format | AT\$TTGSMJDC=? |
| Response | \$TTGSMJDC: (0) (1),(1-255),(1-255),(1-900) (2),(1-31),(0-99),(0,300) OK |
| Write Format | AT\$TTGSMJDC=<enable>,<Signal threshold>,<C1 threshold>[,<period>] |
| Response | OK |
| Read Format | AT\$TTGSMJDC? |
| Response | \$TTGSMJDC: "<enable>,<Signal threshold>,<C1 threshold>,<period>" |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <enable> | Disable/enable the Jamming detection function 0 = disable (default value) 1 = enable 2 = enhanced mode |
| <Signal threshold> | <enable> = 1: Jamming detection algorithm used to compare the difference of two signal strength detects. Range: 1-255 <enable> = 2: He threshold of signal quality (CSQ value) Range : 1- 31 |

| | |
|----------------|---|
| | Default: 10 |
| <C1 threshold> | <p><enabled> = 1: Jamming detection algorithm used to compare the difference of two C1 value detects Range: 1 -255</p> <p><enabled> = 2: Jamming cell number threshold Range: 0-99</p> <p>Default: 40</p> |
| <period> | <p><enable> = 1: Time in seconds to define the jamming detection period.</p> <p><enable> = 2: When the device detects the jamming, the device based on <period> parameter to trigger the enter/quit jamming event</p> <p>The default value is 30.</p> |
| Notes | N/A |

Function Commands

AT\$TTCAIDLST Call ID List Definition

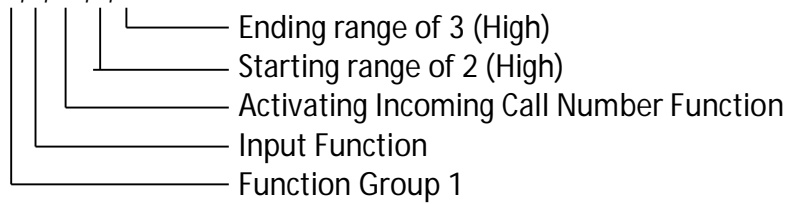
| | |
|------------------|---|
| AT\$TTCAIDLST | Call ID list definition |
| Command Function | This command allows the user to define up to 5 separate incoming call number user input functions |
| Group | |
| Query Format | AT\$TTCAIDLST=? |
| Response | \$TTCAIDLST: (0-5),(0-2), 44 OK |
| Write Format | AT\$TTCAIDLST=<entry>,<mode>[, <"number">] |
| Response | OK |
| Read Format | AT\$TTCAIDLST? |
| Response | \$TTCAIDLST: <entry>,<mode>,<"number"> |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <entry> | 1-5 = Selects which CID entry to modify |
| <mode> | 0 = Disable function generation for incoming call number 1 = Enable function generation for incoming call number and suppress ring indication and respond to network with busy signal. 2 = Enable function generation for incoming call number and do not suppress ring indication. |
| <number> | string type; Character string [-]<0..9,+,>?. Where <?> is a single character wildcard. If number starts with '~' it will match to any incoming call number with 0 or more digits preceding the remaining digits in the string. This is useful for matching to local, national and international ISDN telephony numbering plans. |

| | |
|----------|---|
| Notes | In the function the incoming call number matches more than one incoming call number selection, the mode selection will be based on priority order. The priority order will be for entries 1 through 5 with entry 1 having the highest priority. |
| Example: | These commands will cause the example in AT\$TTFNT to trigger for incoming call numbers matching function call id 2 or function call id 3. |

AT\$CAIDLST=2,1,"987654321?" // Define Incoming call number with the last digit a wildcard

AT\$CAIDLST=2,1,"~987654321" // Define Incoming call number for local and international prefixes

AT\$TTFNT=1,1,65,2,3



AT\$TTFNT User Defined Input/Output

| | |
|------------------|---|
| T\$TTFNT | User Defined Input / Output |
| Command Function | This command allows the user to customize the device input and output capabilities. Any combination of input functions can be monitored to trigger any combination of output functions. |
| Query Format | AT\$TTFNT=? |
| Response | \$TTFNT: (0-99),(0-6),(0-255),(-2147483647 - 2147483647),(-2147483647 - 2147483647) |
| Write Format | AT\$TTFNT=<function group>,<function type>,<function category>,<parm1>,<parm2> |
| Response | OK |
| Read Format | AT\$TTFNT? |
| Response | \$TTFNT: |

| | fntgp | fntyp | fntcat | p1 | p2 |
|--|---|-------|--------|----|----|
| | 1A | 0 | 19 | 1 | 1 |
| | 1B | 3 | 8 | 1 | 1 |
| | 2A | 0 | 19 | 0 | 0 |
| | 2B | 3 | 8 | 0 | 0 |
| | 3A | 0 | 11 | 2 | 4 |
| | 3B | 3 | 18 | 1 | 0 |
| | 4A | 0 | 11 | 5 | 5 |
| | 4B | 4 | 11 | 1 | 1 |
| | 4C | 3 | 7 | 1 | 1 |
| | 5A | 0 | 11 | 0 | 0 |
| | 5B | 3 | 7 | 0 | 0 |
| Execution Format | N/A | | | | |
| Response | | | | | |
| Parameter Values | | | | | |
| <function group> | <p>This parameter defines the group number of a group of functions and the order they are executed. Functions are grouped together to control execution sequence. A group number has to have at least one input function and one output function. Multiple input functions within a group number would be treated as a logical AND condition. Multiple output functions within a group number would be executed individually in a sequential manner.</p> <p>Valid values for group number are: 1 thru 99.</p> <p>For additional details see the Function Tables section of this manual.</p> | | | | |
| <function type> | <p>This parameter defines the type of function: Input or Output. An Input function can be defined as: Transition, Occurrence, or Input. The output function is executed when input function conditions are met.</p> <p>For additional details see the Function Tables section of this manual.</p> | | | | |
| <function category> , <parm1>, <parm2> | <p>These parameters define the actual Input or Output Function number and their valid range for <parm1> and <parm2>.</p> | | | | |

| | |
|-------|--|
| | For additional details see the Function Tables section of this manual. |
| Notes | A maximum of 400 functions (input and output) are supported. User variable values are automatically saved to flash and explicit user intervention to save the values is not required. AT&W does not save user variables to flash. At the end of each function engine cycle, if any user variable has been changed the user variable values are automatically saved to flash at that time (so you can make multiple changes to user variables inside function groups without worrying about writing to flash too often). Changing a user variable via \$AT\$TTRGEV also causes the user variable values to be saved to flash. |

AT\$TTFNTDEL Delete Function

| | |
|------------------|--|
| AT\$TTFNTDEL | Delete Function |
| Command Function | This command allows the user to delete items from the user generated function table. Entering only the group number will delete the whole group. |
| Query Format | N/A |
| Response | N/A |
| Write Format | N/A |
| Response | N/A |
| Read Format | N/A |
| Response | N/A |
| Execution Format | AT\$TTFNTDEL=<group>[<letter ID>] |
| Response | OK |
| Parameter Values | |
| <group> | function list group number |
| <letter ID> | letter indicating which element of the group (optional), it is case sensitive [a..z,A..Z] |
| Notes | |

| | |
|----------|--|
| Example: | AT\$TTFNTDEL=1 Will delete all entries function group 1 AT\$TTFNTDEL=1B Will delete only the second entry in function group 1 |
|----------|--|

AT\$TTFNTDELA Delete Function (All)

| | |
|------------------|---|
| AT\$TTFNTDELA | Delete Function (All) |
| Command Function | This command allows the user to delete all functions from the function table. |
| Query Format | N/A |
| Response | N/A |
| Write Format | N/A |
| Response | N/A |
| Read Format | N/A |
| Response | N/A |
| Execution Format | AT\$TTFNTDELA |
| Response | OK |
| Parameter Values | N/A |
| Notes | N/A |

AT\$TTFNTDELR Delete a Range of Function Groups

| | |
|------------------|--|
| AT\$TTFNTDELR | Delete a Range of Function Groups |
| Command Function | This command deletes a range of function groups. |
| Query Format | AT\$TTFNTDELR=? |
| Response | \$FNTDELR: (0-99),(9-99) OK |
| Write Format | AT\$TTFNTDELR=<start>,<stop> |
| Response | OK |

| | |
|------------------|--|
| Read Format | N/A |
| Response | N/A |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <start> | First group index in range to be deleted |
| <stop> | Last group index in range to be deleted. |
| Notes | N/A |

AT\$TTFNTGQRY Function Group Query

| | |
|------------------|--|
| AT\$TTFNTGQRY | Function Group Query |
| Command Function | Allows user to query the contents of a single function group (same output format as \$TTFNT?, but limited to a single function group). |
| Query Format | AT\$TTFNTGQRY=? |
| Response | AT\$TTFNTGQRY: (0-99) OK |
| Write Format | N/A |
| Response | N/A |
| Read Format | N/A |
| Response | N/A |
| Execution Format | AT\$TTFNTGQRY=<function group> |
| Response | \$TTFNTGQRY: fntgp fnttyp fntcat p1 p2 <pre> 1A 0 27 1 1 1B 3 22 0 0 </pre> OK |
| Parameter Values | |
| <function group> | This parameter defines the group number. Valid values for group number are: 1 thru 99. |
| Notes | None. |

AT\$TTTRGEV Trigger (Test) Input Function

| | |
|------------------|--|
| AT\$TTTRGEV | Trigger Input Function |
| Command Function | This command allows the user to generate any input function. This is useful for testing the user function table. |
| Query Format | AT\$TTTRGEV=? |
| Response | \$TTTRGEV: (0-49),(-2147483648-2147483647),(-32768-32767) OK |
| Write Format | AT\$TTTRGEV=<function>,<state>[,<index>] |
| Response | OK |
| Read Format | N/A |
| Response | N/A |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <function> | Input function number |
| <state> | Input function test state |
| <index> | It is the index of the input function number, it will be applied only for input functions with index in its definition. |
| Notes | User variable values are automatically saved to flash and explicit user intervention to save the values is not required. Changing a user variable via AT\$TTTRGEV also causes the user variable values to be saved to flash. |

AT\$TTCNT User Defined Input Function Counter

| | |
|------------------|--|
| AT\$TTCNT | User Defined Input Function Counter |
| Command Function | This command allows the user to define up to 32 separate counter |
| Query Format | AT\$TTCNT=? |

| | |
|------------------|--|
| Response | \$TTCNT: (1-32), (1-8) OK |
| Write Format | AT\$TTCNT=<index>, <counter type>[, <limit>[, <group>]] |
| Response | OK |
| Read Format | AT\$TTCNT? |
| Response | \$TTCNT: <index>, <counter type>, <limit>, <group> |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <index> | Number of the counter |
| <counter type> | Type of counter 1 – Distance – measure distance when enabled. 2 – Time – measure time when enabled. 3 – High Time – measure time when Function is active. 4 – Low Time – measure time when Function is not active. 5 – High Distance – measure distance when Function is active. 6 – Low Distance – measure distance when Function is not active. 7 – Accumulator – increment via actions. 8 – Toggle Accumulator – increments on every change to set for given <group>. |
| <limit> | This is the limit of the counter, it will be used if in the function definition is not included the limit. If the limit is equal 0, the counter will be disabled. |
| <group> | This is the group number defined in the AT\$TTFNT, it will be used only for counter type 3, 4, 5, 6 and 8 |
| Notes | N/A |
| Example | Define counter number 7 to be incremented by actions AT\$TTCNT=1,7,15 |

AT\$TTCNTQRY Query Counter

| | |
|------------------|--|
| AT\$TTCNTQRY | Query the counter |
| Command Function | This command shows the current value of the specific counter indicated by the argument. |
| Query Format | AT\$TTCNTQRY=? |
| Response | \$TTCNTQRY: (1-32) OK |
| Write Format | AT\$TTCNTQRY=<index> |
| Response | \$TTCNTQRY:<index>, <counter type>, <limit>, <count> OK |
| Read Format | AT\$TTCNTQRY? |
| Response | ERROR |
| Execution Format | AT\$TTCNTQRY=10 |
| Response | \$TTCNTQRY: 10, 1, 0.000 OK |
| Parameter Values | |
| <index> | Number of the counter. If this value is 0, the modem shows all counter |
| <counter type> | Type of counter 1 – Distance – measure distance when enabled. 2 - Time – measure time when enabled. 3 - High Time – measure time when Function is active. 4 - Low Time – measure time when Function is not active. 5 - High Distance – measure distance when Function is active. 6 - Low Distance – measure distance when Function is not active. 7 - Accumulator – increment via actions. 8 – Toggle Accumulator – increments on every change to set for given indicator. |
| <limit> | This is the limit of the counter, it will be used if in the function definition is not included the limit |

| | |
|---------|------------------------------|
| <Count> | Current value of the counter |
| Notes | N/A |

AT\$TTFNTQRY Query the State or Value of the Specified Input Function

| | |
|----------------------|--|
| AT\$TTFNTQRY | Query the State or Value of the Specified Input Function |
| Command Function | This command allows the user to query the state or value of the input function number |
| Query Format | AT\$TTFNTQRY=? |
| Response | \$TTFNTQRY: (0 - <max input function>) OK |
| Write Format | N/A |
| Response | N/A |
| Read Format | N/A |
| Response | N/A |
| Execution Format | AT\$TTFNTQRY=<input function> |
| Response | \$TTFNTQRY: <input function> = state |
| Parameter Values | |
| <max input function> | Maximum input function # supported by the current firmware build (see \$TTFNT for the current maximum input function value). |
| <input function> | Range: 0-<max input function> Selects which input function to query |
| Notes | N/A |
| Example: | AT\$TTFNTQRY=29 /* query input function 29 (Motion function) */ \$TTFNTQRY: 29=0 /* Not motion*/ OK |

AT\$TTSTOCMD Store AT Command Functions

| | |
|------------------|---|
| AT\$TTSTOCMD | Store AT Command Functions |
| Command Function | This command allows the user to store AT command output functions. The AT command is executed upon the triggering of the associated input function. |
| Query Format | AT\$TTSTOCMD=? |
| Response | \$TTSTOCMD: (1-35),<AT commands> OK |
| Write Format | AT\$TTSTOCMD = <1-35>, <AT command> |
| Response | OK |
| Read Format | AT\$TTSTOCMD? |
| Response | \$TTSTOCMD: AT Function# AT Cnds 1 2 35 OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <1-35> | AT function index. |
| <AT command> | AT command associated with the AT function index. The AT commands is not checked for validity (maximum 240 bytes) |
| Notes | <p>This command is used in conjunction with the Dynamic Input Output function (AT\$TTFNT). The output function associated with this command is function 28. When output function 28 is defined in the function table, Parm1 defines which index to refer to.</p> <p>The AT command associated with the index is executed. The use of Dynamic Function Scripting using AT\$TTFNT or AT\$TTFNTDEL as a stored AT Command Function can lead to</p> |

| | |
|--|--|
| | <p>unpredictable operation and is not recommended. When storing command to dial a voice call, a "v" replaces the ";" at the end of the dial string..(i.e., atd17195551212v)</p> <p>It is possible to use concatenate commands.</p> |
|--|--|

Miscellaneous Commands

AT\$TTUSRVAL User Values definition

| | |
|------------------|--|
| AT\$TTUSRVAL | User variable |
| Command Function | This command allows the user to capture a value in a temporarily memory which can be later capture |
| Query Format | AT\$TTUSRVAL=? |
| Response | \$TTUSRVAL: (User Value) OK |
| Write Format | AT\$TTUSRVAL=<User Value> |
| Response | OK |
| Read Format | AT\$TTUSRVAL? |
| Response | \$TTUSRVAL:<Hex Value> |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <User Value> | The user variable to be set. It is a hex value in the range 0 to FFFFFFFF |
| Notes | |

AT\$TTICCID Integrated Circuit Card ID

| | |
|------------------|---|
| AT\$TTICCID | Integrated Circuit Card ID |
| Command Function | This command allows the user to retrieve the Integrated Circuit Card Identification (ICCID) from the SIM. |
| Query Format | AT\$TTICCID=? |
| Response | OK |

| | |
|------------------|---|
| Write Format | N/A |
| Response | OK |
| Read Format | AT\$TTICCID? |
| Response | \$TTICCID: <iccid> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | N/A |
| Notes | A SIM must be inserted into the module and initialized in order to read the ICCID of the SIM (GSM 11.11 Chapter 10.1.1) |

AT%DM Debug Mode*

| | |
|---------------------|---|
| AT%DM | Debug Mode |
| Command Function | This command allows the user to view debug mode operation, including GPS and GSM internal operation |
| Query Format | AT%DM=? |
| Response | %DM: (0-3),(0-3) OK |
| Write Format | AT%DM=<Event debug level>,<GSM debug mode>,<GPS debug mode> |
| Response | OK |
| Read Format | AT%DM? |
| Response | %DM: <Event debug level>,<GSM debug mode>,<GPS debug mode> |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <Event debug level> | 0 = No debug 1 = Show all messages |

| | |
|------------------|--|
| <GSM debug mode> | 0= No debug 1 =Basic GSM debug 2= GSM and GPRS debug |
| <GPS debug mode> | 0= No debug 1: Basic GPS debug 3= Detailed GPS debug |
| Notes | <p>All debug message should have the time when those ones where generated</p> <p>The definition of each level is:</p> <p><Event debug level> : Show messages related with all events, when the input function type is true, when the output function is executed, parse of the command executed</p> <p>0 = No debug 1= Show all messages</p> <p><GSM debug mode> : Show the messages related with the GSM and GPRS registration, all changes in the GSM state will be monitoring, depending the debug level setup</p> <p>0= No debug 1 =Basic GSM debug Include, changes in the status of the GSM network (at+creg), changes in the signal quality, sim pin needed, sms received, voice call init and received</p> <p>2= GSM and GPRS debug Include, changes in the status of the GPRS network (at%cgreg), changes in IP, messages sent to the server (all option, messages sent, messages save in the buffer, errors), messages received from the server by GPRS</p> <p><GPS debug mode>: Show the messages related with the GPS, all changes in the GPS status can be monitoring.</p> <p>0= No debug 1: Basic GPS debug Include when the device get GPS, when the unit lost GPS, when the GPS is reset</p> <p>3= Detailed GPS debug Include the information received from the GPS receiver</p> |

| | |
|--|--|
| | * In development and subject to change |
|--|--|

AT\$TTDEVSTAT Query Device Status*

| | |
|------------------|--|
| AT\$TTMDSTAT | Query Device Status |
| Command Function | This command allows the user to display many modem status parameters with a single command. |
| Query Format | AT\$TTMDSTAT=? |
| Response | OK |
| Write Format | N/A |
| Response | N/A |
| Read Format | AT\$TTDEVSTAT? |
| Response | <pre> \$DEVTYP: 0,9142 \$PKG: 50 OEM_MODEM +CPIN: READY HPLMN: 310,410 +CREG: 1, CB90, 290F %CGREG: 1, CB90, 290F +COPS: 310,410 ARFCN: 198 BAND: 850 +CSQ: 14,0 +CGACT: 1,0 \$NETIP: "000.000.000.000", "000.000.000.000", "000.000.000.000" DREG_NW: 0001 DREG_ME: 0005 DT_CAUSE: CAFF DT_TYPE: 0004 \$LUPREJ: 0,0 \$RAUREJ: 0,0 +CEER: 0,1,1,255,no error \$CGEER: no PDP reject cause TCP ERROR: 0 PAD DISC: 0 API DISC: 5 OK </pre> |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <\$DEVTYP> | see AT\$TTDEVSTYP? command |

| | |
|------------|--|
| <\$PKG> | see AT\$TTPKG command |
| <+CPIN> | see AT+CPIN? command |
| <HPLMN> | Country code and Network code from SIM |
| <+CREG> | see AT+CREG? command |
| <%CGREG> | see AT%CGREG? command |
| <+COPS> | see AT+COPS? command (numeric only) |
| <+CSQ> | see AT+CSQ command |
| <+CGACT> | see AT+CGACT? command |
| <\$NETIP> | see AT\$TTNETIP? command |
| <DREG_NW> | DREG_NW values (Deregistration - Network) FFFF = not set 0 = no service 1 = limited service |
| <DREG_ME> | DREG_ME values (Deregistration - ME) 0 = not set 1 = GPRS detach 2 = IMSI detach 3 = Combined detach 4 = SIM removed 5 = Power off 6 = GPRS disabled 7 = Limited service 8 = Soft power off |
| <DT_CAUSE> | see table in notes section below |
| <DT_TYPE> | DT_TYPE (Detach type) 1 = GPRS detach 2 = IMSI detach 3 = Combined detach 4 = SIM removed 5 = Power off 6 = GPRS disabled 7 = Limited service 8 = Soft power off |

| | |
|-------------|--|
| <\$LUPREJ> | see AT\$TTLUPREG? command |
| <\$RAUREJ> | see AT\$TTRAUREG? command |
| <+CEER> | see AT+CEER command |
| <TCP ERROR> | <p>TCP ERROR values</p> <p>0 = no error stored</p> <p>1 = normal closure</p> <p>2 = unexpected RST</p> <p>3 = unexpected SYN</p> <p>4 = received FIN</p> <p>5 = timeout establishing connection</p> <p>6 = failure establishing connection</p> <p>7 = timeout in close</p> <p>8 = timeout on established connection</p> <p>9 = RST on established connection</p> <p>10 = ICMP rec'd: no route to host</p> <p>11 = ICMP rec'd: connection refused</p> <p>12 = ICMP rec'd: frag req, DF set</p> <p>13 = no route found for dest</p> <p>14 = interface down or other failure</p> <p>15 = internal stack failure</p> |
| <PAD DISC> | <p>0 = no reason stored</p> <p>1 = server timeout</p> <p>2 = PPP caused</p> <p>3 = host caused</p> <p>4 = connection timeout</p> <p>5 = caused by AT cmd</p> |
| <API DISC> | <p>0 = no reason stored</p> <p>4 = connection timeout</p> <p>5 = caused by AT cmd</p> <p>6 = bad msg format</p> <p>7 = connect fail</p> |
| Notes | * In development and subject to change |

AT\$TTSRN Serial Number

| | |
|------------------|---|
| AT\$TTSRN | Read the serial number of the device |
| Command Function | This command allows the user to query the serial number if the device |
| Query Format | AT\$TTSRN=? |
| Response | OK |
| Write Format | N/A |
| Response | N/A |
| Read Format | AT\$TTSRN |
| Response | \$TTSRN:<sn> |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <sn> | The serial number of the device |
| Notes | N/A |

AT\$RESET Reset device

| | |
|------------------|---|
| AT\$RESET | Reset the device |
| Command Function | This command allows the user to reboot the device |
| Query Format | AT\$RESET=? |
| Response | OK |
| Write Format | N/A |
| Response | N/A |
| Read Format | AT\$RESET |
| Response | OK <rebooting the device ...> |
| Execution Format | N/A |
| Response | N/A |

| | |
|------------------|---|
| Parameter Values | |
| Notes | The rebooting will take several seconds because it should detach network and close GPS and save the current GPS position before rebooting |

AT\$TTPWSRV Password for Server and APN

| | |
|------------------------------|---|
| AT\$TTPWSRV | Set authorization for setup the APN and Server destination |
| Command Function | This command allows the user to enable or disable authorization for setup the APN and Server destination |
| Command Format Query | AT\$TTPWSRV=? |
| Response | AT\$TTPWSRV: <"oldpasswd">,<mask "newpasswd"> OK |
| Write Format | AT\$TTPWSRV=<"oldpasswd">,<"newpasswd"> |
| Response | OK |
| Read Format | N/A |
| Response | N/A |
| Execution Format Response | N/A |
| Response | N/A |
| Parameter Values | |
| <oldpasswd> | Specified when the password is being changed. |
| <newpasswd> | This is the value of the new password and is specified only when the password is being changed. It must be no more than eight characters in length and must be enclosed in double quotes. |
| <passwd> | Specified when changing the authorization mask for setup the APN and Server destination |
| <mask> | Bit mask specifying which command will be authorized. Each bit specifies one interface as enumerated in the table below. |

| | Bit value | Interface | Command |
|-------|--|--------------------|--------------|
| | 1 | APN | At+cgdcont |
| | 2 | Destination Server | AT\$TTSRVDST |
| | To select multiple items to authorize, add the bit values of each interface to be authorized. | | |
| Notes | It is possible setup the mask without setup password. Before to remove the password it is necessary remove the mask | | |

AT\$TTMAPLINK Google Map Link

| | |
|------------------|--|
| AT\$TTMAPLINK | Send SMS with Google maps hyperlink |
| Command Function | This command allows the user to receive a SMS with Google maps hyperlink if the user calls the device. |
| Query Format | AT\$TTMAPLINK=? |
| Response | \$TTMAPLINK: ("Phone number") OK |
| Write Format | AT\$TTMAPLINK=<phone number> |
| Response | OK |
| Read Format | N/A |
| Response | N/A |
| Execution Format | AT\$TTMAPLINK |
| Response | OK |
| Parameter Values | |
| <phone number> | The destination phone number. It is an optional parameter, if it is specified the Google map link will be sent to this phone number. |
| Notes | Execute format can be used only via SMS channel |

AT\$TTMAPLINKACT Control Map Link function

| | |
|------------------|---|
| AT\$TTMAPLINKACT | Control the map link function |
| Command Function | Enables or disables the Map Link function. |
| Query Format | AT\$TTMAPLINKACT=? |
| Response | \$TTMAPLINKACT: (0,1) OK |
| Write Format | AT\$TTMAPLINKACT=<state> |
| Response | OK |
| Read Format | AT\$TTMAPLINKACT? |
| Response | \$TTMAPLINKACT: <state> |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <state> | 0: Disable map link function 1: Enable map link function (default) |
| Notes | |

AT\$TTFNTEXE Execute Output Function

| | |
|----------------------|---|
| AT\$TTFNTEXE | Execute the specified output event directly |
| Command functions: | This command allows the user to execute an output function event immediately. |
| Command Format Query | AT\$TTFNTEXE=? |
| Response | \$TTFNTEXE: (0-58) ,(0-2147483647),(-2147483648-2147483647), (0,1) OK |
| Write Format | AT\$TTFNTEXE= <function>,<param1>,<param2>,<show ok> |
| Response | OK |

| | |
|------------------|---|
| Read Format | N/A |
| Response | N/A |
| Parameter values | |
| <function> | output function category |
| <param1> | param1 for the output function |
| <param2> | param2 for the output function. |
| <show ok> | A numeric to decide whether to output "OK" response. The default value is 1. 0: do not output "OK" response 1: output "OK" response |
| Notes | |

AT\$TTRSTCFG Reset Method Configuration

| | |
|------------------|---|
| AT\$TTRSTCFG | Reset method configuration |
| Command Function | This command allows the user to define which method the device will use to reset itself. This includes resets caused by commands, output functions, network monitoring or by the ACK timeout. |
| Query Format | AT\$TTRSTCFG =? |
| Response | \$TTRSTCFG: (0-2),(0-1) OK |
| Write Format | AT\$TTRSTCFG= <reset method>[,<need dereg>] |
| Response | OK |
| Read Format | AT\$TTRSTCFG? |
| Response | \$TTRSTCFG: <reset method>,<need dereg> |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| < reset method > | 0: Assert (default) 1: MCU reboot |

| | |
|--------------|---|
| | 2: System reboot. |
| <need dereg> | 0: Don't de-register from network before resetting 1: De-register from network before resetting. (default) |
| Notes | N/A |
| Examples | |

AT+ESMLCK SIM Lock Cell Carrier Registration

| | |
|------------------|--|
| AT+ESMLCK | Cell Carrier Registration for SIM Lock |
| Command Function | This command allows the user to clear or add carriers to be allowed on a SIM lock. |
| Query Format | AT+ESMLCK=? |
| Response | +ESMLCK:(0-4), (0-4), <key>, <data_imsi>, <data_gid1>, <data_gid2> OK |
| Write Format | AT+ESMLCK= <category>,<operation>,<key>,<data_imsi>[,<data_gid1>, <data_gid2>] |
| Response | OK |
| Read Format | AT+ESMLCK? |
| Response | +ESMLCK: (0,2,2,3,0,0,10),(1,2,3,0,0,30,0),(2,2,3,0,0,30,0),(3,2,3,0,0,30,0),(4,2,3,0,0,1,0),(5,2,3,0,0,1,0),(6,2,3,0,0,10,1),"1040080002053742",0,"255",0,"255" ,2 |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <category> | A numeric value to indicate the lock category. It should be 0 always. |
| <operation> | A numeric value to indicate the operation type of the command to do. 2: Add a network which can be allowed to be working with 3: Clear the network which can be allowed to be working with |

| | |
|-------------|---|
| <key> | The PIN code of the SIM card. |
| <data_imsi> | It's the MCC/MNC. <MCC/MNC>: Mobile Country Code and Mobile Network Code. <ul style="list-style-type: none"> ● MCC: Mobile country code. It is 3 digits in length and ranges from 000-999. ● MNC: Mobile network code. It is 3 digits in length and ranges from 000-999. |

AT+CLCK Facility Lock

| | |
|----------------------|--|
| AT+CLCK | Facility lock |
| Command functions | This command allows the user to lock and unlock some facilities. |
| Command Format Query | AT+CLCK=? |
| Response | +CLCK: (list of supported <fac>) OK |
| Write Format | AT+CLCK = <fac>,<mode>[,<passwd>] |
| Response | If <mode><>2 and Command is successful OK If <mode>=2 (for query) and Command is successful +CLCK: <status> [<CR><LF> +CLCK: <status>....] OK |
| Read Format | N/A |
| Response | N/A |
| Execute Format | N/A |
| Response | N/A |
| Parameters | |
| <fac> | "PS" PH-SIM (lock Phone to SIM card) (ME asks password when other than current SIM card inserted; ME may remember certain amount of previously used cards thus not requiring password when they are inserted) "SC" SIM (lock SIM card) (SIM asks password in ME power-up and when this lock command issued) "FD" SIM fixed dialing memory: If the mobile is locked to "FD", only the phone numbers stored to the "FD" memory can |

| | |
|----------|---|
| | be dialed "PF" Lock Phone to the very first SIM card "PN" Network Personalization (refer GSM 02.22) |
| <mode> | 0 Unlock 1 Lock 2 Query status |
| <passwd> | Password in a string with length of 4 – 8. |

GPS Commands

AT\$TTGPSSTT GPS Command

| | |
|------------------|--|
| AT\$TTGPSSTT | GPS Status |
| Command Function | This command allows the user to configure the state of the GPS module. |
| Query Format | AT\$TTGPSSTT=? |
| Response | \$TTGPSSTT: (0-3) OK |
| Write Format | AT\$TTGPSSTT=<cmd> |
| Response | OK |
| Read Format | AT\$TTGPSSTT? |
| Response | \$TTGPSSTT: <cmd> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <cmd> | 0 – Disables the GPS 1 – Cold start 2 – Hot start 3 – Stop the current GPS fix |
| Notes | 0 - GPS is in an idle state 1 - Commands GPS to begin acquisition from a cold start. 2 - Hot start can only be accomplished if it has valid time, ephemeris, and almanac and position data. 3 - Some Skypatrol products do not tracking GPS, and is defined as GPS fix stop (for power consumption) |

AT\$TTGPSPORT - Configure Sending of GPS Data to the Serial Port

| AT\$TTGPSPORT | Configure Sending of GPS data to the Serial Port | | | | | | | | | | |
|------------------|---|-----------------|----------------------|---|-----|---|-----|---|-----|----|-----|
| Command Function | This command allows the user to configure sending of GPS data on the serial port. | | | | | | | | | | |
| Query Format | AT\$TTGPSPORT=? | | | | | | | | | | |
| Response | \$TTGPSPORT: (0-1),(0-127) OK | | | | | | | | | | |
| Write Format | AT\$TTGPSPORT=<option>,<nmeaMsgs> | | | | | | | | | | |
| Response | OK | | | | | | | | | | |
| Read Format | AT\$TTGPSPORT? | | | | | | | | | | |
| Response | \$TTGPSPORT: <option>,<nmeaMsgs> OK | | | | | | | | | | |
| Execution Format | N/A | | | | | | | | | | |
| Response | N/A | | | | | | | | | | |
| Parameter Values | | | | | | | | | | | |
| <option> | 0 – Disable sending of GPS data to the local port when the device is in AT command mode (Default) 1 – Enable sending of GPS NMEA ASCII data to the local port when the device is in AT command mode | | | | | | | | | | |
| <nmeaMsgs> | This field is the bit-wise OR of the type of messages desired. The user has following message options to select from. Decimal Format <table border="1" data-bbox="573 1461 1192 1745"> <thead> <tr> <th>User Selectable</th> <th>Type of NMEA Message</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>GGA</td> </tr> <tr> <td>4</td> <td>GSA</td> </tr> <tr> <td>8</td> <td>GSV</td> </tr> <tr> <td>16</td> <td>RMC</td> </tr> </tbody> </table> | User Selectable | Type of NMEA Message | 1 | GGA | 4 | GSA | 8 | GSV | 16 | RMC |
| User Selectable | Type of NMEA Message | | | | | | | | | | |
| 1 | GGA | | | | | | | | | | |
| 4 | GSA | | | | | | | | | | |
| 8 | GSV | | | | | | | | | | |
| 16 | RMC | | | | | | | | | | |
| Notes | It is used for debug purpose as directed by Skypatrol Technical Support Personnel. | | | | | | | | | | |

AT\$TTGPSQRY Read Current GPS ASCII data

| AT\$TTGPSQRY | Read Current GPS ASCII data | | | | | | | | | | |
|------------------|---|-----------------|----------------------|------|-----|------|-----|------|-----|------|-----|
| Command Function | This command allows a user to read current NMEA format GPS data. | | | | | | | | | | |
| Query Format | AT\$TTGPSQRY=? | | | | | | | | | | |
| Response | \$TTGPSQRY: [(0-7F),(0-127)],(0-1) OK | | | | | | | | | | |
| Write Format | N/A | | | | | | | | | | |
| Response | N/A | | | | | | | | | | |
| Read Format | AT\$TTGPSQRY=<nmeaMsgs>,<decimal> | | | | | | | | | | |
| Response | "\$TTGPSQRY....." OK | | | | | | | | | | |
| Execution Format | N/A | | | | | | | | | | |
| Response_ | N/A | | | | | | | | | | |
| Parameter Values | The output NMEA sentence depends on whether the <nmeaMsgs> parameter is entered in Hex or Decimal format. By default, the <decimal> parameter is not required and <nmeaMsgs> parameter has to be entered as HEX value without the preceding "0x" characters as outlined in Hex Format table below. | | | | | | | | | | |
| <nmeaMsgs> | <p>This field is the sum of the type of NMEA messages desired. A user has the following message options to select from. Maximum value for <nmeaMsgs> in this case would be 7F in Hex format or 127 in decimal format.</p> <p>Hex Format</p> <table border="1"> <thead> <tr> <th>User Selectable</th> <th>Type of NMEA Message</th> </tr> </thead> <tbody> <tr> <td>0x01</td> <td>GGA</td> </tr> <tr> <td>0x04</td> <td>GSA</td> </tr> <tr> <td>0x08</td> <td>GSV</td> </tr> <tr> <td>0x10</td> <td>RMC</td> </tr> </tbody> </table> | User Selectable | Type of NMEA Message | 0x01 | GGA | 0x04 | GSA | 0x08 | GSV | 0x10 | RMC |
| User Selectable | Type of NMEA Message | | | | | | | | | | |
| 0x01 | GGA | | | | | | | | | | |
| 0x04 | GSA | | | | | | | | | | |
| 0x08 | GSV | | | | | | | | | | |
| 0x10 | RMC | | | | | | | | | | |

| | Decimal Format <table border="1"> <thead> <tr> <th>User Selectable</th> <th>Type of NMEA Message</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>GGA</td> </tr> <tr> <td>4</td> <td>GSA</td> </tr> <tr> <td>8</td> <td>GSV</td> </tr> <tr> <td>16</td> <td>RMC</td> </tr> </tbody> </table> | User Selectable | Type of NMEA Message | 1 | GGA | 4 | GSA | 8 | GSV | 16 | RMC |
|-----------------|--|-----------------|----------------------|---|-----|---|-----|---|-----|----|-----|
| User Selectable | Type of NMEA Message | | | | | | | | | | |
| 1 | GGA | | | | | | | | | | |
| 4 | GSA | | | | | | | | | | |
| 8 | GSV | | | | | | | | | | |
| 16 | RMC | | | | | | | | | | |
| <decimal> | 1 = <nmeaMsg> value has to be sum of User Selectable values from decimal table format 0 = select values out of hex table format | | | | | | | | | | |
| Notes | N/A | | | | | | | | | | |

AT\$TTGPSRI Set and Query GPS Reset Interval

| | |
|------------------|--|
| AT\$TTGPSRI | Set and Query GPS Reset Interval |
| Command Function | <p>This command allows the user to specify the duration of a continuous GPS no lock condition after which the GPS chip will be reset.</p> <p>The GPS will often take several minutes to get a lock when power has been cycled to the device. However, if valid RTC time is available, then the lock should be acquired more quickly. The second parameter allows the user to cut down the time without a lock before resetting the GPS chip when outside of the cold start window.</p> |
| Query Format | AT\$TTGPSRI=? |
| Response | \$TTGPSRI: (0-255),(0-255) OK |
| Write Format | AT\$TTGPSRI=<coldStartInterval>,<warmStartInterval> |
| Response | OK |
| Read Format | AT\$TTGPSRI? |
| Response | \$TTGPSRI: <coldStartInterval>,<warmStartInterval> |

| | |
|---------------------|--|
| | OK |
| Execution Format | AT\$TTGPSRI |
| Response | ERROR |
| Parameter Values | |
| <coldStartInterval> | When RTC time is not valid (year <2010), this is the time (in tens of minutes) of GPS outage after which to reset the GPS chip and resume positioning. |
| <warmStartInterval> | When RTC time is valid (year >= 2010) and this value is zero, then coldStartInterval is used as defined above. When RTC time is valid (year >= 2010) and this value is greater than zero, this is the time (in minutes) of GPS outage after which to reset the GPS chip and resume positioning. |
| Notes | If <coldStartInterval> is set to 0, the GPS chip will not be reset after a GPS outage of any duration. |
| Example | AT\$TTGPSRI=1,2 Code will allow 10 minutes to acquire a GPS lock before resetting the GPS chip following a power cycle (RTC time not valid). Once a lock has been acquired (RTC time automatically set when a lock has been acquired), the code will allow two minutes of 'no GPS lock' time before resetting the GPS chip. Since RTC time is maintained through an ignition reset, the two minute limit would be in effect following an ignition reset. |

AT\$TTODOM GPS Trip Odometer

| | |
|------------------|--|
| AT\$TTODOM | GPS Odometer |
| Command Function | The \$ODOM command records how far the vehicle has traveled total. |
| Query Format | AT\$TTODOM=? |
| Response | \$TTODOM: (0-2147483647) |
| Write Format | N/A |
| Response | OK |
| Read Format | AT\$TTODOM? |
| Response | \$TTODOM xxxx (xxxx=distance traveled in meters) |
| Execution Format | N/A |

| | |
|------------------|---|
| Response | N/A |
| Parameter Values | N/A |
| Notes | It is not possible reset the odometer value to 0. |

AT\$TTTODOM Virtual Trip Odometer

| | |
|------------------|--|
| AT\$TTTODOM | Virtual Trip Odometer |
| Command Function | The \$TTTODOM command records how far the vehicle has traveled. This is identical to the \$TTODOM feature, but allows the user to reset this trip odometer without resetting the other odometer. |
| Query Format | AT\$TTTODOM=? |
| Response | \$TTTODOM:(0-2147483647) OK |
| Write Format | AT\$TTTODOM=<new odometer> |
| Response | OK |
| Read Format | AT\$TTTODOM? |
| Response | \$TTTODOM: xxxx (xxxx=distance traveled in meters) |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | N/A |
| <new odometer> | New trip odometer, distances defined in meters |
| Notes | <p>The user is able to set a initial value for the Trip Odometer starting at 0 but not higher than the maximum value of 2147483647.</p> <p>The AT&F command will not reset the odometer value to 0. (see example below)</p> <p>The Trip Odometer reading would be a 4-byte value starting from 0 to 4000000000 (maximum of approximately 2500000 miles before it rolls over to 0)</p> <p>The Trip Odometer shall be in meters.</p> <p>The \$TTTODOM value can trigger the function engine via input function 26.</p> |

| | |
|---------|--|
| Example | Reset Trip Odometer to 0: AT\$TTTODOM=0 |
|---------|--|

AT\$TTODOENA Enable odometer with ignition OFF

| | |
|------------------|--|
| AT\$TTODOENA | Enable odometer with ignition OFF |
| Command Function | This command allow to the used enable or disable odometer when ignition is OFF. |
| Query Format | AT\$TTODOENA =? |
| Response | \$TTODOENA: (0-1) OK |
| Write Format | AT\$TTODOENA =<odo_enable> |
| Response | OK |
| Read Format | AT\$TTODOENA? |
| Response | AT\$TTODOENA:<odo_enable> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <odo_enable> | 0 = disable odometer with ignition off. 1 = enable odometer with ignition off (default) |
| Notes | Disable odometer means, the odometer will be not incremented. |
| Example | |

AT\$TTOVSDEF Define and Query GPS Over speed Interval

| | |
|--------------|--|
| AT\$TTOVSDEF | Define and Query GPS Over speed Interval |
|--------------|--|

| | |
|------------------|--|
| Command Function | This command allows the user to define the criteria for a GPS over speed function. A GPS over speed function occurs when the minimum speed that is defined by the <speed> parameter is maintained for a specific duration of time. |
| Query Format | AT\$TTOVSDEF=? |
| Response | \$TTOVSDEF: (1-3), (0 – 65535),(0-65535) OK |
| Write Format | AT\$TTOVSDEF=<index>, <speed>,<interval> |
| Response | OK |
| Read Format | AT\$TTOVSDEF? |
| Response | \$TTOVSDEF: <index>, <speed>,<interval>,<status>,<max_speed>,<duration> OK |
| Execution Format | AT\$TTOVSDEF |
| Response | ERROR |
| Parameter Values | |
| <index> | 1 – 3 This is the position of the over speed definition. |
| <speed> | Speed, in knots, must be met and/or exceeded to trigger the GPS over speed function. |
| <interval> | Number of consecutive seconds for which <speed> must be maintained to trigger the GPS over speed function. |
| <status> | If 1, then <max_speed> and <duration> represent a GPS over speed interval that is currently active. If 0, they represent the previous GPS over speed interval. |
| <max_speed> | The highest speed (in knots) that was attained in the current or previous GPS over speed interval. |
| <duration> | Number of consecutive seconds that the speed was at or above <speed>. |
| Notes | If <speed> is set to zero, the GPS over speed function is disabled. |

AT\$TTGEOFNC Geo Fencing a Circular Area

| | |
|------------------|---|
| AT\$TTGEOFNC | Geo Fencing a Circular Area |
| Command Function | This command allows a user to define one or group of circular geo fencing. (Maximum 10 circular geofencing per group) |
| Query Format | AT\$TTGEOFNC=? |
| Response | \$TTGEOFNC: (1-25),(0-1000000),(-90.0 - +90.0),(-180.0 - +180.0)[..... (-90.0 - +90.0),(-180.0 - +180.0)] OK |
| Write Format | AT\$TTGEOFNC=<index> <radius>,<latitude1>,<longitude1>[..... <latitude10>,<longitude10>] |
| Response | OK |
| Read Format | AT\$TTGEOFNC? |
| Response | OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <index> | Defines the circular geofence index (single or group) |
| <radius> | Defines radius of the circle from given Latitude and Longitude coordinates (in meters) |
| <latitudex> | Defines the latitude for the center point of a circle |
| <longitudex> | Defines the longitude for the center point of a circle |
| Notes | N/A |

AT\$TTGFIND Circular Geofence Query

| | |
|------------------|--|
| AT\$TTGFIND | Circular GeoFence Query |
| Command Function | This command allows a user to query the value(s) assigned to specific index storage for the Circular GeoFence. |
| Query Format | AT\$TTGFIND=? |

| | |
|------------------|---|
| Response | \$TTGFIND:(1 - 25) OK |
| Execution Format | AT\$TTGFIND=<index> |
| Response | Fencenum: <index> Radius: <radius> NO Lat Lng 1 <latitude1> <longitude1> 2 <latitude2> <longitude2> 10 <latitude10> <longitude10> OK |
| Parameter Values | |
| <Index> | Defines the circular geofence index (single or group) |
| <radius> | Defines radius of the circle from given Latitude and Longitude coordinates (in meters) |
| <latitude> | Defines the latitude for the center point of a circle |
| <longitude> | Defines the longitude for the center point of a circle |
| Notes | N/A |

AT\$TTGFDB Set Geofence Debounce Count

| | |
|------------------|--|
| AT\$TTGFDB | Set Geofence Debounce Count |
| Command Function | This command allows the user to set the number of consecutive geofence positions required to trigger an 'inside geofence' or 'outside geofence' function. It prevents a false reading when the device is on the threshold of the geofence. This applies for all type of geo fences |
| Query Format | AT\$TTGFDB=? |
| Response | \$GFDB: (0-250),(0-250) OK |

| | |
|------------------|---|
| Write Format | AT\$TTGFDB=<out_cnt>,<in_cnt> |
| Response | OK |
| Read Format | AT\$TTGFDB? |
| Response | \$GFDB: <out_cnt>, <in_cnt> OK |
| Execution Format | N/A |
| Response | |
| Parameter Values | |
| <out_cnt> | Consecutive GPS position reports outside a geofence required to trigger '0' condition for geofence input function (see \$TTFNT) |
| <in_cnt> | Consecutive GPS position reports inside a geofence required to trigger '1' condition for geofence input function (see \$TTFNT) |
| Notes | The GPS reporting interval is set every one second. |

AT\$TTGFDEL Delete a Range of Geo-Fences

| | |
|------------------|--|
| AT\$TTGFDEL | Delete a Range of Geo-Fences |
| Command Function | This command deletes a range of geo-fences. |
| Query Format | AT\$TTGFDEL=? |
| Response | \$TTGFDEL: (1-25),(1-25) OK |
| Write Format | AT\$TTGFDEL=<start>,<stop> |
| Response | OK |
| Read Format | N/A |
| Response | N/A |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <start> | First geo-fence index in range to be deleted |
| <stop> | Last geo-fence index in range to be deleted. |

| | |
|-------|--|
| Notes | To delete one group of the circular geo fence, <start> and <stop> must have the same value |
|-------|--|

AT\$TTPLYGFN Geo Fencing a Polygonal Area

| | |
|------------------|--|
| AT\$TTPLYGFN | Geo Fencing a Polygonal Area |
| Command Function | This command allows a user to define up to 25 separate polygonal geographical areas. The function engine (see AT\$TTFNT) can be set up to send a message when the device moves in or out of the geographical area defined by the polygon. Each command defines up to 10 possible vertices or corners. The line segments of the polygon are generated by connecting the non-zero vertices in sequence (for example, point 0 is connected to point 1, which is connected to point 2, etc.). The polygon can be generated in either a clockwise or counterclockwise sequence (see Notes section below for additional details). You can specify your polygon with less than 10 vertices, but the first and last points of the polygon should be identical. |
| Query Format | AT\$TTPLYGFN=? |
| Response | \$TTPLYGFN: (1-25),(-90.0 - +90.0),(-180.0 - +180.0)[..... (-90.0 - +90.0),(-180.0 - +180.0)] OK |
| Write Format | AT\$TTPLYGFN=<index>,<latitude1>,<longitude1>.....,<latitude10>,<longitude10> |
| Response | OK |
| Read Format | AT\$TTPLYGFN? |
| Response | OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <index> | 1-25: index for the geographic polygon area |
| <latitude> | Defines the latitude for each vertices of the polygon in decimal degrees (for example, 32 degrees 30 minutes 0 seconds would be represented as 32.5 degrees since 30 minutes is exactly 1/2 of a degree). The latitude must be specified in 15 characters |

| | |
|-------------|--|
| | (including minus sign and decimal point) or less. This is the same representation for latitude as used in the AT\$TTGEOFNC command. |
| <longitude> | Defines the longitude for each vertices of the polygon in decimal degrees (for example, -96 degrees 45 minutes 0 seconds would be represented as -96.75 degrees since 45 minutes is exactly 3/4 of a degree). The longitude must be specified in 15 characters (including minus sign and decimal point) or less. This is the same representation for longitude as used in the AT\$TTGEOFNC command. |
| Notes | <p>Verify the integrity of your polygon description with a mapping tool. The most common mistake with polygons is to mistype one of the fractional digits in one of the coordinates. When this happens, the polygon you've created with your mapping tool is not the polygon you've entered into this command and you will not get your desired results.</p> <p>Regardless of how many points you specify for your polygon, all 10 points are fed to the algorithm. Due to the mathematical nature of this algorithm, the polygon will not work if vertices of the same polygon lie on opposite sides of the 180 degree longitude divide.</p> <p>The debounce defined to the geofence (AT\$TTGFNDB) will be use to determine if the device is inside of the polygon or outside of the polygon.</p> |
| Example | <p>The following commands create a crude diamond-shaped polygon #9:</p> <pre>AT\$TTPLYGFN=9,25.801446,-80.333126,25.804933,- 80.332718,25.804817,-80.3304,25.803175,-80.3304,25.80333,- 80.328577,25.801359,-80.328469,25.801446,-80.333126</pre> |

AT\$TTPLYGFIND Polygonal Geo Fencing Query

| | |
|------------------|--|
| AT\$TTPLYGFIND | Polygonal Geo Fencing Query |
| Command Function | This command allows a user to query the device for the value(s) assigned to specific index storage for the Polygonal Geo Fence |
| Query Format | AT\$TTPLYGFIND=? |
| Response | \$TTPLYGFIND:(1 - 25) |

| | |
|------------------|--|
| | OK |
| Execution Format | AT\$TTPLYGFIND=<index> |
| Response | Fencenum: <index> Radius: <radius> NO Lat Lng 1 <latitude1> <longitude1> 2 <latitude2> <longitude2> 10 <latitude10> <longitude10> OK |
| Parameter Values | |
| <Index> | Defines the polygonal geofence index |
| <latitude> | Defines the latitude for each vertex defined for the polygon |
| <longitude> | Defines the longitude for each vertex defined for the polygon |
| Notes | N/A |

AT\$TTPLYDEL Delete Polygonal Geofence

| | |
|------------------|---|
| AT\$TTPLYDEL | Delete Polygonal Geofence |
| Command Function | This command allows a user to delete all the points for the indicated polygonal geofence (see \$TTPLYFN). |
| Query Format | AT\$TTPLYDEL=? |
| Response | \$TTPLYDEL: (1-25) OK |
| Write Format | AT\$TTPLYDEL=<index> |
| Response | OK |
| Read Format | N/A |
| Response | N/A |
| Execution Format | N/A |
| Response | N/A |

| | |
|------------------|--|
| Parameter Values | |
| <index> | 1-25: index for the geographic polygon area (see \$TTPLYGFN) |
| Notes | None. |

AT\$TTRROUTE Geo Fencing a Route

| | |
|------------------|---|
| AT\$TTRROUTE | Geo Fencing a Route |
| Command Function | <p>This command allows a user to define up to 25 separate route. The function engine (see AT\$TTFNT) can be set up to send a message when the device moves in or out of the route.</p> <p>The line segments of the route are generated by connecting the non-zero points in sequence (for example, point 0 is connected to point 1, which is connected to point 2, etc.). The route can be generated in either direction (see Notes section below for additional details). You can specify your route with less than 10 points.</p> |
| Query Format | AT\$TTRROUTE=? |
| Response | <p>\$TTRROUTE:(1-25),(0-1000),(-90.0 - +90.0),(-180.0 - +180.0)[..... (-90.0 - +90.0),(-180.0 - +180.0)]</p> <p>OK</p> |
| Write Format | AT\$TTRROUTE=<index>,<range>,<latitude1>,<longitude1>,,,,,,,<latitude10>,<longitude10> |
| Response | OK |
| Read Format | AT\$TTRROUT? |
| Response | OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <index> | 1-25: index for the route |
| <range> | The range meters from the coordinates linear path to be consider fenced |
| <latitude> | Defines the latitude for each point of the route in decimal degrees (for example, 32 degrees 30 minutes 0 seconds would |

| | |
|-------------|--|
| | be represented as 32.5 degrees since 30 minutes is exactly 1/2 of a degree). The latitude must be specified in 15 characters (including minus sign and decimal point) or less. This is the same representation for latitude as used in the AT\$TTGEOFNC command. |
| <longitude> | Defines the longitude for each point of the route in decimal degrees (for example, -96 degrees 45 minutes 0 seconds would be represented as -96.75 degrees since 45 minutes is exactly 3/4 of a degree). The longitude must be specified in 15 characters (including minus sign and decimal point) or less. This is the same representation for longitude as used in the AT\$TTGEOFNC command. |
| Notes | The debounce defined to the geofence (AT\$TTGFNDB) will be use to determine if the device is inside the route or outside the route. |
| Example | The following commands create a route #9: AT\$TTROUTE=9,300,25.778060,-80.359730,25.772220,- 80.330560,25.769000,-80352050,25.776110,- 80.369540,25.782780,-80.36954096.. OK |

AT\$TTROUTEIND ROUTE Geo Fencing Query

| | |
|------------------|---|
| AT\$TTROUTEIND | Route Geo Fencing Query |
| Command Function | This command allows a user to query the device for the value(s) assigned to specific index storage for the Route GeoFence |
| Query Format | AT\$TTROUTEIND=? |
| Response | \$TTROUTEIND:(1 - 25) OK |
| Execution Format | AT\$TTROUTEIND=<index> |
| Response | Fencenum: <index> Range: <range> NO Lat Lng 1 <latitude1> <longitude1> 2 <latitude2> <longitude2> |

| | |
|------------------|---|
| | 10 <latitude10> <longitude10> OK |
| Parameter Values | |
| <Index> | Defines the route geofence index |
| <range> | The range meters from the coordinates linear path to be consider fenced (in meters) |
| <latitude> | Defines the latitude for each point of the route |
| <longitude> | Defines the longitude for each point of the route |
| Notes | N/A |

AT\$TTROUTEDEL Delete Route Geofence

| | |
|------------------|---|
| AT\$TTROUTEDEL | Delete Route Geofence |
| Command Function | This command allows a user to delete all the points for the indicated polygonal geofence (see \$TTROUTE). |
| Query Format | AT\$TTROUTEDEL=? |
| Response | \$TTROUTEDEL: (1-25) OK |
| Write Format | AT\$TTROUTEDEL=<index> |
| Response | OK |
| Read Format | N/A |
| Response | N/A |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <index> | 1-25: Defines the route geofence index (see \$TTROUTE) |
| Notes | None. |

AT\$TTGPSHEAD GPS Heading Change

| | |
|--------------------|--|
| AT\$TTGPSHEAD | Set an angle threshold to monitor GPS heading change |
| Command Function | This command set an angle threshold to monitor GPS heading change. This command should be used together with the input event 51. |
| Query Format | AT\$TTGPSHEAD =? |
| Response | \$TTGPSHEAD: (1-359), (0-120) OK |
| Write Format | AT\$TTGPSHEAD = <change threshold>,<debounce> |
| Response | OK |
| Read Format | AT\$TTGPSHEAD? |
| Response | \$TTGPSHEAD:0,0 OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <change threshold> | The angle threshold. If the heading change is beyond the threshold, the unit will think it is turning event. Range: 0 - 359 |
| <debounce> | The continuous time, in seconds, before the unit think it is turning. |
| Notes | None. |

Motion Commands

The optional features described in this section are not applicable to all devices. For information about which features are present for each product, please refer to the product specification sheet.

AT\$TTACCAM Motion Configuration

| | |
|------------------|--|
| AT\$TTACCAM | Motion configuration command |
| Command Function | This command allows the user to set the motion parameters of the internal sensor. |
| Query Format | AT\$TTACCAM=? |
| Response | \$TTACCAM: (0-1),(0-10),(1-255),(1-10) OK |
| Write Format | \$TTACCAM=<enable>,<threshold>,<reset duration>,<motion duration> |
| Response | OK |
| Read Format | AT\$TTACCAM? |
| Response | \$TTACCAM:<enable>,<threshold>,<reset duration>,<motion duration> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <enable> | Disable/enable the any motion feature 0 = disable (default value) 1 = enable |
| <threshold> | The threshold is the accelerometer sensor used to measure the changes of acceleration, if the change of acceleration is larger than it, accelerometer sensor will start rest duration count. Otherwise, the accelerometer sensor will start motion duration count. |

| | |
|-------------------|---|
| | Range: 2-10 Default: 3 |
| <reset duration> | Time to determine whether the device enters rest status, i.e. Device will enter rest status if the accelerometer sensor starts rest duration count and maintains for a period of time defined by the parameter <Rest Duration>. Range: 1 - 255(*15s) Default: 3 |
| <motion duration> | Time to determine whether the device enters motion status, i.e. Device will enter motion status if the accelerometer sensor starts motion duration count and maintains for a period of time defined by the parameter <Motion Duration>. Range: 1 - 10 (*100ms) Default: 4 |
| Notes | N/A |
| Examples | |

Buzzer Commands

The optional features described in this section are not applicable to all devices. For information about which features are present for each product, please refer to the product specification sheet.

AT\$TTBUZCFG Buzzer Configuration

| | |
|------------------|---|
| AT\$TTBUZCFG | Buzzer configuration command |
| Command Function | This command allows the user to configure the frequency and duty for BUZZER. The two parameters will affect the sound of the BUZZER |
| Query Format | AT\$TTBUZCFG=? |
| Response | \$TTBUZCFG: (1-5000),(0-100) OK |
| Write Format | AT\$TTBUZCFG=<freq>,<duty> |
| Response | OK |
| Read Format | AT\$TTBUZCFG? |
| Response | \$TTBUZCFG: <freq>,<duty> OK |
| Execute Format | N/A |
| Response | N/A |
| Parameter values | |
| <freq> | The frequency of the square wave shape to drive the internal buzzer. If it is lower than 200, the buzzer can not work normally. |
| <duty> | The duty of the square wave shape to drive the internal buzzer. If it is equal to 0 or 100, the buzzer can not work normally. |
| Notes | N/A |
| Examples | |

AT\$TTBUZCTR Buzzer Control

| | |
|------------------|---|
| AT\$TTBUZCTR | Buzzer control command |
| Command Function | This command allows the user to control when to start and when to stop the BUZZER |
| Query Format | AT\$TTBUZCTR=? |
| Response | \$ TTBUZCTR: (0-1) OK |
| Write Format | AT\$TTBUZCTR=<state> |
| Response | OK |
| Read Format | N/A |
| Response | N/A |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <enable> | A numeric to indicate to start or stop the buzzer. 0: stop the buzzer. 1: start the buzzer. |
| Notes | N/A |
| Examples | |

Macros

Macros are a functionality predefined in the Skypatrol devices, those functionalities allow to the customer easier configuration of the device.

Macros use the counter, variables and groups, which can not be used as a part of other script because will affect the predefined functions.

The script created using those macros can be modified only with the respective command

AT\$TTSP Set protocol

| | |
|------------------|--|
| AT\$TTSP | Set protocol |
| Command Function | Define the output function used to send the message to the server or serial port when the macro functions are used |
| Query Format | AT\$TTSP=? |
| Response | \$TTSP: (20-26,37,38) OK |
| Write Format | AT\$TTSP=<outputfnc> |
| Response | OK |
| Read Format | AT\$TTSP? |
| Response | \$TTSP:<outputfnc> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <outputfnc> | Output function to be used, default value 26 |
| Notes | |

AT\$TTDI Set Distance Interval

| | |
|------------------|---|
| AT\$TTDI | Set distance interval |
| Command Function | The unit will generate the position report when the distance is reached |
| Query Format | AT\$TTDI=? |
| Response | \$TTDI: (0-65535) OK |
| Write Format | AT\$TTDI=<distance> |
| Response | OK |
| Read Format | AT\$TTDI? |
| Response | \$TTDI:<distance> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <distance> | Distance interval |
| Notes | |

AT\$TTDL Dump Log Records

| | |
|------------------|-------------------------------------|
| AT\$TTDL | Dump Log Records |
| Command Function | Resend the records saved in the log |
| Query Format | AT\$TTDL=? |
| Response | \$TTDL: (0-65535) OK |
| Write Format | N/A |
| Response | N/A |
| Read Format | N/A |
| Response | N/A |

| | |
|------------------|-----------------------------|
| Execution Format | AT\$TTDL=<numrec> |
| Response | OK |
| Parameter Values | |
| <numrec> | Number of record to be sent |
| Notes | |

AT\$TTHB Set Heartbeat Period

| | |
|------------------|--|
| AT\$TTHB | Set Heartbeat Period |
| Command Function | The unit will generate a keep alive report according with the time defined as a parameter |
| Query Format | AT\$TTHB=? |
| Response | \$TTHB:(0-16777215)[,"IMEI" "REPORT"] OK |
| Write Format | AT\$TTHB=<interval>[,<msgtype>] |
| Response | OK |
| Read Format | AT\$TTHB? |
| Response | \$TTHB: <interval>,<msgtype> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <interval> | Number of seconds defined for the periodic report |
| <msgtype> | Message type used to when the report is sent. It is an optional parameter, if it is omitted the "IMEI" value must be used as default |
| Notes | |

AT\$TTHC Set Heading Change Warning

| | |
|------------------|--|
| AT\$TTHC | Set Heading Change Warning |
| Command Function | The unit will generate a position report when the heading change for a period defined |
| Query Format | AT\$TTHC=? |
| Response | \$TTHC:(0-359),(0-255) OK |
| Write Format | AT\$TTHC=<heading>,<period> |
| Response | OK |
| Read Format | AT\$TTHC? |
| Response | \$TTHC:<heading>,<period> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <heading> | Set the degrees of the heading |
| <period> | Time in seconds to defined the period in which the heading must be before to send the report |
| Notes | |

AT\$TTIPHB Set Heartbeat when IP Change

| | |
|------------------|--|
| AT\$TTIPHB | Set Heartbeat when IP Change |
| Command Function | The unit will generate a report when the device get a new IP |
| Query Format | AT\$TTIPHB=? |
| Response | \$TTIPHB:["IMEI" "REPORT" "DEL"] OK |
| Write Format | AT\$TTIPHB=[<msgtype>] |
| Response | OK |

| | |
|------------------|--|
| Read Format | N/A |
| Response | N/A |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <msgtype> | Message type used to when the report is sent. It is an optional parameter, if it is omitted the "IMEI" value must be used as default |
| Notes | |

AT\$TTIW Set Idle Warning

| | |
|------------------|---|
| AT\$TTIW | Set Idle Warning |
| Command Function | The unit will generate a position report when the device is in the same position for the time defined and the input conditions happen |
| Query Format | AT\$TTIW=? |
| Response | \$TTIW: (0-1000000),(0-2),(0-2) OK |
| Write Format | AT\$TTIW=<period>,<ignsatate>,<iostate> |
| Response | OK |
| Read Format | AT\$TTIW? |
| Response | \$TTIW:<period>,<ignsatate>,<iostate> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <period> | Set the number of second that motion is not detected before to send the position report |
| <ignstate> | State of the ignition |
| <iostate> | State of the IO1 |

| | |
|-------|--|
| Notes | |
|-------|--|

AT\$TTNR Set Network Restart Period

| | |
|------------------|--|
| AT\$TTNR | Set Network restart period |
| Command Function | Set the time before the IP session is closed and restarted |
| Query Format | AT\$TTNR=? |
| Response | \$TTNR: (0-16777215),(0-1) OK |
| Write Format | AT\$TTNR=<period><sndrpt> |
| Response | OK |
| Read Format | AT\$TTNR? |
| Response | \$TTNR:<period><sndrpt> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <period> | Set the number of second before the IP session is shutdown and restated |
| <sndrpt> | Define if one position report is generated of not. It is an optional parameter if is omitted the default value must be 0 |
| Notes | |

AT\$TTRI Set Report Interval

| | |
|------------------|--|
| AT\$TTRI | Set Report Interval |
| Command Function | Set the position report interval when the device is moving over the velocity defined |
| Query Format | AT\$TTRI=? |
| Response | \$TTRI: (0-16777215), (0-16777215),(0-255), (0-16777215) |

| | |
|------------------|---|
| | OK |
| Write Format | AT\$TTRI=<period>,<rst_period>,<velocity> <vel_period> |
| Response | OK |
| Read Format | AT\$TTRI? |
| Response | \$TTRI:<period>,<rst_period>,<velocity>, <vel_period> |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <period> | Set the position report interval when the velocity is above the optional parameter <velocity> |
| <rst_period> | Set the period reset interval |
| <velocity> | Velocity |
| <vel_period> | Set the position report interval when the velocity is below the optional parameter <velocity> |
| Notes | |

AT\$TTRN Report Current Position

| | |
|------------------|--|
| AT\$TTRN | Report current position |
| Command Function | The unit will generate a current position report, using the format mask defined. |
| Query Format | AT\$TTRN=? |
| Response | OK |
| Write Format | N/A |
| Response | N/A |
| Read Format | N/A |
| Response | N/A |
| Execution Format | AT\$TTRN |
| Response | OK |
| Parameter Values | |

| | |
|-------|--|
| Notes | |
|-------|--|

AT\$TTRR Set Report After Reset

| | |
|------------------|---|
| AT\$TTRR | Set Report after Reset |
| Command Function | Define the behavior of the device after the reset process |
| Query Format | AT\$TTRR=? |
| Response | \$TTRR: (0-16), (0-16777215),(0-16777215) OK |
| Write Format | AT\$TTRR: <numrpt>,<gpswait>,<period> |
| Response | OK |
| Read Format | AT\$TTRR? |
| Response | \$TTRR: <numrpt>,<gpswait>,<period> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <numrpt> | Number of reports send after the reset (default 1) |
| <gpswait> | Time in second waited for the device to get the GPS |
| <period> | Set the interval in seconds to be used by the device to send the <numrpt> |
| Notes | |

AT\$TTSI Set Interrupt

| | |
|------------------|---|
| AT\$TTSI | Set interrupt |
| Command Function | Set Interrupts thresholds, debounce and hysteresis |
| Query Format | AT\$TTSI=? |
| Response | \$TTSI: (s b g1 g2),(0-255),(0-16777215), , (0-255) OK |

| | |
|------------------|--|
| Write Format | AT\$TTSI=<inttype>,<range>,<debounce>,<hysteresis> |
| Response | OK |
| Read Format | AT\$TTSI? |
| Response | \$TTSI: s,<range>,<debounce>,<hysteresis> b,<range>,<debounce>,<hysteresis> g1,<range>,<debounce>,<hysteresis> g2,<range>,<debounce>,<hysteresis> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <inttype> | Type of interrupt to be set: s: Velocity b: External battery voltage g1: input1 g2: input2 |
| <range> | Values within the range specified for <inttype> s: Velocity (mph) b: External battery voltage (volts) g1: 0: disabled 1: Interrupt on any transition 2: Interrupt on low to high transition 3: Interrupt on high to low transition g2: input2 0: disabled 1: Interrupt on any transition 2: Interrupt on low to high transition 3: Interrupt on high to low transition |
| <debounce> | Set time in seconds that signal need to be stable before to generate a position report |
| Notes | |

AT\$TTSR Set output state

| | |
|------------------|--|
| AT\$TTSR | Set output state |
| Command Function | Set output1 to high or low |
| Query Format | AT\$TTSR=? |
| Response | \$TTSI: (0-2) OK |
| Write Format | N/A |
| Response | N/A |
| Read Format | AT\$TTSR? |
| Response | \$TTSR: <state> OK |
| Execution Format | AT\$TTSR: <state> |
| Response | OK |
| Parameter Values | |
| <state> | State of the output 0: Set low 1: Set high 2: Set high only when IP channel is active |
| Notes | |

AT\$TTTA Set Tow Alert Warning

| | |
|------------------|-------------------------|
| AT\$TTTA | Set Tow Alert Warning |
| Command Function | Set towing alert |
| Query Format | AT\$TTTA=? |
| Response | \$TTTA: (0-65535) OK |
| Write Format | AT\$TTTA=<period> |
| Response | OK |

| | |
|------------------|--|
| Read Format | AT\$TTTA? |
| Response | \$TTTA: <period> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <period> | Set time in seconds for detection of motion when ignition is OFF |
| Notes | |

AT\$TTTE Set Trip event

| | |
|------------------|--|
| AT\$TTTE | Set Trip event |
| Command Function | Set the parameter to generate the trip event messages |
| Query Format | AT\$TTTE=? |
| Response | \$TTTE: (0-255), (0,65535) OK |
| Write Format | AT\$TTTE=<velocity>,<time> |
| Response | OK |
| Read Format | AT\$TTTE? |
| Response | \$TTTE: <velocity>,<time> OK |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <velocity> | Set the minimum velocity required to star a trip. |
| <time> | Set time in seconds to wait with velocity below to <velocity> before to end trip |
| Notes | |

AT\$TTRBI Set Report Interval based on ignition

| | |
|------------------|--|
| AT\$TTRBI | Set Report Interval based on ignition |
| Command Function | Set the position report interval when the device is moving over the velocity defined |
| Query Format | AT\$TTRBI=? |
| Response | \$TTRBI: (0-16777215), (0-16777215) OK |
| Write Format | AT\$TTRI=<ign_on_period>,<ign_off_period> |
| Response | OK |
| Read Format | AT\$TTRBI? |
| Response | \$TTRBI:<ign_on_period>,<ign_off_period> |
| Execution Format | N/A |
| Response | N/A |
| Parameter Values | |
| <ign_on_period> | Set the position report period when the ignition is ON |
| <ign_off_period> | Set the position report period when the ignition is OFF |
| Notes | |

Groups and counters

This chapter describes the groups, counters, variables and geo fences used for each command

| Command | Description | Groups | Counter | Variable | Geo | Stored Cmd | Param1 |
|------------|------------------------------|---------------------------------|----------------|----------|-----|------------|--------|
| AT\$TTDI | Set distance interval | 200 | 32 | | | | 11 |
| AT\$TTDL | Dump Log Records | | | | | | |
| AT\$TTHB | Set Heartbeat Period | 201 | 31 | | | | 11 |
| AT\$TTHC | Set Heading Change Warning | 202 | | | | | 21 |
| AT\$TTIPHB | Set Heartbeat when IP Change | 204 | | | | | 33 |
| AT\$TTIW | Set Idle Warning | 205 245 | 16 | | | | 17 |
| AT\$TTNR | Set Network Restart Period | 206 207 | 29 | | | | 24 |
| AT\$TTRI | Set Report Interval | 213 214 215 216 | 26 27 28 | | | | 10 |
| AT\$TTRN | Report Current Position | 217 | | | | | 16 |
| AT\$TTRR | Set Report After Reset | 218 219 220 221 222 | 23 24 25 | 0 | | | 1 |



TT8750+ AT Command Set

| | | | | | | | |
|-----------|---------------------------------------|-----|----|--|----|----|----|
| AT\$TTSI | Set Interrupt | 223 | 21 | | | | 2 |
| | | 224 | 22 | | | | 9 |
| | | 225 | | | | | 4 |
| | | 226 | | | | | 5 |
| | | 227 | | | | | 6 |
| | | 228 | | | | | 7 |
| | | 229 | | | | | |
| | | 230 | | | | | |
| AT\$TTSR | Set output state | 232 | | | | 29 | |
| | | 233 | | | | 30 | |
| | | | | | | 31 | |
| AT\$TTTA | Set Tow Alert warning | 234 | 20 | | 24 | | 20 |
| | | 235 | | | | | |
| | | 236 | | | | | |
| AT\$TTTE | Set Trip event | 237 | 19 | | | | 25 |
| | | 238 | | | | | 32 |
| | | 239 | | | | | |
| AT\$TTRBI | Set Report Interval based on ignition | 240 | 17 | | | | 10 |
| | | 241 | 18 | | | | |
| | | 242 | | | | | |

Function Tables

Function Type

The <function type> parameter defines the type of function: Input or Output. An Input function can be defined as: Transition, Occurrence, or Input. The Output function is executed when input function conditions are met.

| Value | Type of function | Description |
|-------|----------------------------------|--|
| 0 | Transition Trigger (Low to High) | A transition Trigger is defined as an input condition, defined by <function category>, whose value was previously <parm1> or less is now greater than <parm1> and less than <parm2> or was greater or equal to <parm2> is now less than <parm2> but greater than <parm1>. The output function would be executed when an input <function category> requirements are satisfied or transition to the value set by <parm1> and <parm2> when they are equal. <parm1> should be the min value and <parm2> should be the max value. |
| | | <p>Example 1:</p>  <p>An output function will be executed when the value of an input function exceeds <Parm1> (previously it was <Parm1> or less) or decreases to a value less than <Parm2> (previously it was <Parm2> or greater).</p> |
| | | <p>Example 2:</p>  <p>An output function will be executed when the value of an input function is 0 (previously it was anything else but 0) and <Parm1> along with <Parm2> is set to 0.</p> |
| | | Example 3: |

| | | |
|---|--------------------|---|
| | | <p>An output function will be executed when the value of an input function is 1 (previously it was anything else but 1) and <Parm1> along with <Parm2> is set to 1.</p> |
| 1 | Occurrence Trigger | <p>An Occurrence Trigger is defined as an input condition, defined by <function category>, whose current value is greater than or equal to <parm1> and less than or equal to <parm2>.</p> <p>The output function would be executed when an input <function category> requirements are satisfied or transition to the value set by <parm1> and <parm2> when they are equal. <parm1> should be the min value and <parm2> should be the max value.</p> |
| | | <p>Example 4:</p> <p>Figure 4. An output function will be executed when the current value of an input function is between <Parm1> and <Parm2> including boundary conditions.</p> |
| | | <p>Example 5:</p> <p>Figure 5. An output function will be executed when the value of the input function changes from 0 to 1 or vice-versa.</p> |
| | | <p>Example 6:</p> <p>Figure 6. An output function will be executed when the value of the input function is 1 and <Parm1> along with <Parm2> is set to 1.</p> |
| 2 | AND condition | <p>AND condition is defined as an input condition, defined by <function category>, that should be used as a logical AND</p> |

| | | |
|---|----------------------------------|---|
| | | <p>condition to another input condition defined as Transition Trigger or an Occurrence Trigger.</p> <p>The Output function will be executed when the combination of the all Input Trigger condition are valid.</p> |
| 3 | Output | An Output function is executed when all input function conditions (defined as Transition Trigger, Occurrence Trigger, or Input Trigger) for that particular <function group> are met. |
| 4 | OR condition | <p>Or condition is defined as an input condition, defined by <function category>, that should be used as a logical OR condition to another input condition defined as Transition Trigger or an Occurrence Trigger.</p> <p>The Output function will be executed when one of the all Input Trigger condition are valid.</p> |
| 5 | NOT condition | NOT condition is defined as an input condition, defined by <function category>, that should be used as a logical NOT condition to another input condition defined as Transition Trigger or an Occurrence Trigger. |
| 6 | Transition Trigger (High to Low) | <p>A transition Trigger is defined as an input condition, defined by <function category>, whose value was greater than <parm1> and less than <parm2> and now is less than <parm1> or gather than <parm2>.</p> <p>The output function would be executed when an input <function category> requirements are satisfied or transition to the value set by <parm1> and <parm2> when they are equal. <parm1> should be the min value and <parm2> should be the max value.</p> |

Function Category

The <function category> parameter defines the actual Input or Output Function number and their valid range for <parm1> and <parm2>.

Input Function Table

The following table defines the values for <function category>, <parm1> and <parm2> parameter for input functions defined as a Transition Trigger, Occurrence Trigger, Input AND condition Trigger, Input OR condition Trigger, Input NOT condition Trigger

| Input Function Table | | | |
|----------------------|--------|--------|--|
| Function Category | Parm1 | Parm2 | Description |
| 0 | 0 or 1 | 0 or 1 | GPIO1 – General purpose Input #1 0 = Low 1 = High |
| 1 | 0 or 1 | 0 or 1 | GPIO2 – General purpose Input #2 |
| 2 | 0 or 1 | 0 or 1 | Reserved |
| 3 | 0 or 1 | 0 or 1 | GPIO4 – General purpose input, detects the state of main power. 0 = Main power disconnected 1 = Main power connected |
| 4 | 0 or 1 | 0 or 1 | GPIO10 is multiplexed with Analog input2. After AT\$TTIOMOD=0 is executed, the PIN will work as GPIO10. |
| 5 | 0 | 0 | Reserved |
| 6 | 0 | 0 | Reserved |
| 7 | 0 | 0 | Reserved |
| 8 | 0 | 0 | Reserved |
| 9 | 0 or 1 | 0 or 1 | GPIO9 – General purpose input, detects the state of Ignition |
| 10 | 1 | 1 | Modem power up indication |
| 11 | 0 to 5 | 0 to 5 | Modem GSM registration (see AT+CREG command description for GSM registration status information) |
| 12 | 0 to 8 | 0 to 8 | Modem GPRS registration (see AT%CGREG command description for GPRS registration status information) |

| | | | |
|----|--------------|--------------|---|
| 13 | 0 or 1 | 0 or 1 | Receipt of IP address. 0 = No IP address 1 = Valid IP address obtained |
| 14 | 1-32 | 0-2147483647 | Counter function, <param1> is a <counter index> and <param2> is a counter limit, if <paream2> is equal 0 then the limit will be the value defined with the command AT\$TTCNT |
| 15 | 0 to 250 | 0-250 | Current Velocity (unit of measurement is: Knots) |
| 16 | 0-16000 | 0-16000 | Analog input 1 function (mV) |
| 17 | 0-16000 | 0-16000 | Analog input 2 function (mV) |
| 18 | 1-25 | 0 or 1 | Geofence #. See AT\$TTGEOFNC command for details on setting a circular geo-fence <param1> is a <geofence index> And <param2> 0 = Leaving Geofence area 1 = Entering Geofence area |
| 19 | 0 or 1 | 0 or 1 | GPS Status 0 = Invalid GPS data 1 = Valid GPS data |
| 20 | 1 | 1 | Time of day function |
| 21 | 1 | 1 | Day of the week function (\$ttddow) |
| 22 | 0 to 1000000 | 1000000 | Invalid GPS data for a period of time (unit of measurement is: seconds) |
| 23 | 0 to 1000000 | 1000000 | Device staying Idle in one place (unit of measurement is: seconds) |
| 24 | 0 – 3200 | 0 – 3200 | Number of Unsent Messages (\$msglogrd) |
| 25 | 0 – 100 | 0 – 100 | Memory full percentage (\$msglogrd) |
| 26 | 0-2147483647 | 0-2147483647 | Current \$TTODOM value |
| 27 | 1-3 | 0-1 | <param1> is an <over speed index> And <param2> is 0 = A GPS over speed interval has ended |

| | | | |
|----|------------------------------|------------------------------|--|
| | | | 1 = A GPS over speed interval has begun |
| 28 | 0-2147483647 | 0-2147483647 | Trip odometer (distance in meters) |
| 29 | 0-1 | 0-1 | Motion Function 0: Not motion 1: Motion |
| 30 | -2147483648 to 2147483647 | -2147483648 to 2147483647 | User variable 0 |
| 31 | -2147483648 to 2147483647 | -2147483648 to 2147483647 | User variable 1 |
| 32 | -2147483648 to 2147483647 | -2147483648 to 2147483647 | User variable 2 |
| 33 | -2147483648 to 2147483647 | -2147483648 to 2147483647 | User variable 3 |
| 34 | -2147483648 to 2147483647 | -2147483648 to 2147483647 | User variable 4 |
| 35 | -2147483648 to 2147483647 | -2147483648 to 2147483647 | User variable 5 |
| 36 | -2147483648 to 2147483647 | -2147483648 to 2147483647 | User variable 6 |
| 37 | -2147483648 to 2147483647 | -2147483648 to 2147483647 | User variable 7 |
| 38 | -2147483648 to 2147483647 | -2147483648 to 2147483647 | User variable 8 |
| 39 | -2147483648 to 2147483647 | -2147483648 to 2147483647 | User variable 9 |
| 40 | 0 or 1 | 0 or 1 | GSM_JAMMING_STATE Current state of the GSM jamming detection algorithm. JAMMED = 1 CLEAR = 0 |
| 41 | 0 | 0 | Reserved |
| 42 | 1-25 | 0 or 1 | Polygon Geofence. See \$TTPLYGFN for details on setting a polygonal geofence. <Param1> is a <polygon geofence index> |

| | | | |
|---------|-------------|-------------|---|
| | | | And <param2> is 0 = Leaving Geofence area 1 = Entering Geofence area |
| 43 | 1-25 | 0 or 1 | Route. See \$TTROUTE for details on setting a Route. <Param1> is a <route index> And <param2> is 0 = Leaving route 1 = Entering route |
| 44 | 1 | 1 | This function is triggered at the completion of a FOTA upgrade after the modem has rebooted with the new firmware (old firmware if the upgrade failed). |
| 45 | 0-359 | 0-359 | Heading function |
| 46 | 0-1 | 0-1 | GPS antenna status 0: Antenna disconnected 1: Antenna connected |
| 47 | 0-1 | 0-1 | Simcard status 0: Removed 1: Inserted |
| 48 | 0-100 | 0-100 | Internal battery level (%) |
| 49 | 1-5 | 1-5 | Voice call identification |
| 50 | 300-1200 | 300-1200 | External battery voltage |
| 51 | 1 | 1 | Heading change (\$GPSHEAD) |
| 52-63 | | | Reserved |
| 64 | -255 to 255 | -255 to 255 | Acceleration based on GPS (m/s ²). |
| 65 – 81 | | | Reserved |
| 82 | -110 to 0 | -110 to 0 | GSM signal quality (dBm) |

Output Function Table

The below table defines the values for <function category>, <parm1> and <parm2> parameter for output functions defined as Output.

| Output Function Table | | | |
|-----------------------|----------------------|-------------------------|--|
| Function Category | Parm1 | Parm2 | Description |
| 0 | 0 | 0 | Reserved |
| 1 | 0 | 0 | Reserved |
| 2 | 0 | 0 | Reserved |
| 3 | 0-1 | 0-1 | Set GPIO3's output state |
| 4 | 0-1 | 0-1 | Reserved |
| 5 | 0-1 | 0-1 | Set GPIO5's output state |
| 6 | 0-1 | 0-1 | Set GPIO6's output state |
| 7 | 0-1 | 0-1 | Set GPIO7's output state, USR1 LED |
| 8 | 0-1 | 0-1 | Set GPIO8's output state, USR2 LED |
| 9 | 0-1 | 0-1 | Start/Stop buzzer |
| 10 | See GPIO Flash Table | | Flash buzzer |
| 11 | 0 | 0 | Reserved |
| 12 | 0 | 0 | Reserved |
| 13 | 0 | 0 | Reserved |
| 14 | 0 | 0 | Reserved |
| 15 | See GPIO Flash Table | | Flash GPIO #3 configured as Output |
| 16 | | | Flash GPIO #5 configured as Output |
| 17 | | | Flash GPIO #6 configured as Output |
| 18 | | | Flash GPIO #7 configured as Output |
| 19 | | | Flash GPIO #8 configured as Output |
| 20 | 0 to 2147483647 | See Messages type Table | Generate and transmit one UDP Message to first IP address and port listed in \$TTSRVDST command on Parm1 and |

| | | | |
|----|-----------------|-------------------------|---|
| | | | Parm2 values |
| 21 | | | Generate and transmit one UDP Message to second IP address and port listed in \$TTSRVDST command on Parm1 and Parm2 values |
| 22 | | | Generate and transmit one UDP Message to third IP address and port listed in \$TTSRVDST command on Parm1 and Parm2 values |
| 23 | | | Generate and transmit one UDP Message to forth IP address and port listed in \$TTSRVDST command on Parm1 and Parm2 values |
| 24 | | | Generate and transmit one UDP Message to fifth IP address and port listed in \$TTSRVDST command on Parm1 and Parm2 values |
| 25 | | | Generate and transmit a UDP message with Acknowledge. This message is controlled by \$TTACKCFG command for number of retries sent. This message has to be acknowledged to avoid sending of retries. |
| 26 | | | Generate and transmit one UDP Message to all IP address and port listed in \$TTSRVDST command based on Parm1 and Parm2 values |
| 27 | 1 – 32 | <limit> | Resets the counter specified by Parm1 to the limit specified by Parm2. |
| 28 | 1 – 25 | 0 | Execute AT command stored at index number of the \$TTSTOCMD command. Parm1 identifies the index number. |
| 29 | 0 to 2147483647 | See Messages type Table | Sends data over SMS to All SMS destination addresses configured via \$TTSMSDST command. (For select \$SMSDST entries, see function categories 54-58) |
| 30 | 0 to | See Messages | Sends data over SMS to the first indexed |

| | | | |
|----|--------------------|-------------------------|--|
| | 2147483647 | type Table | SMS destination address configured via \$TTSMSDST command. |
| 31 | 0 to 2147483647 | | Sends data over SMS to the second indexed SMS destination address configured via \$TTSMSDST command. |
| 32 | 0 to 2147483647 | | Sends data over SMS to the third indexed SMS destination address configured via \$TTSMSDST command. |
| 33 | 0 to 2147483647 | | Sends data over SMS to the fourth indexed SMS destination address configured via \$TTSMSDST command. |
| 34 | 0 to 2147483647 | | Sends data over SMS to the fifth indexed SMS destination address configured via \$TTSMSDST command. |
| 35 | 1 – 25 | 0 - 1000000 | Set circular geo-fence specified by parm1 to current latitude & longitude with radius specified by parm2 |
| 36 | 0 – 57 | 0 to –1 | Emulate AT\$TTTRGEV command via function engine. Parm1 is the input function number while Parm2 is the value to emulate for the input function |
| 37 | 0 to –1 | See Messages type Table | Generate and transmit one TCP/IP Message to the first IP address & port number listed by \$TTSRVDST command based on Parm1 and Parm2 values |
| 38 | 0 to -1 | See Messages type Table | Generate and transmit message to main serial port based on Parm1 and Parm2 values in ASCII. If messages format is in binary the information will be show in hexadecimal format |
| 39 | 0 to –1 | See Messages type Table | Save the Message in the log based on Parm1 and Parm2 values |
| 40 | 0-1 | 0-1 | Enter the GPS in sleep mode 0: No sleep 1: Sleep |
| 41 | 0-2 | 0-2 | Enter the Communication module in sleep mode |

| | | | |
|----|-------|---------------------------|--|
| | | | <p>0: No sleep</p> <p>1: GPRS in sleep</p> <p>2: GSM and GRPS in sleep</p> |
| 42 | 0 – 9 | -2147483648 to 2147483647 | Sets user variable indicated by parm1 to value of parm2 (for example, if parm1 is 7 and parm2 is 50, this output function would set user variable 7 to 50) |
| 43 | 0 – 9 | -2147483648 to 2147483647 | Increments user variable indicated by parm1 by value of parm2 (for example, if parm1 is 4, parm2 is 100, and user variable 4 was 200 prior to this function, user variable 4 would be incremented to 300 by this function) |
| 44 | 0-9 | -2147483648 to 2147483647 | Decrements user variable indicated by parm1 by value of parm2 (for example, if parm1 is 4, parm2 is 100, and user variable 4 was 200 prior to this function, user variable 4 would be decremented to 100 by this function) |
| 45 | 0 – 9 | -2147483648 to 2147483647 | Multiply user variable indicated by parm1 by value of parm2 (for example, if parm1 is 9, parm2 is 10, and user variable 9 was 50 prior to this function, user variable 9 would be incremented to 500 by this function) |
| 46 | 0-9 | -2147483648 to 2147483647 | Divide user variable indicated by parm1 by value of parm2 (for example, if parm1 is 4, parm2 is 100, and user variable 4 was 200 prior to this function, user variable 4 would be decremented to 2 by this function) |
| 47 | 0 – 9 | -2147483648 to 2147483647 | Copies value of a system variable into user variable indicated by parm1. Parm2 is used as an index to determine the system variable that will be copied (see User Variable Index Table) |
| 48 | 0-4 | 0-1 | Send buffer <param1> |

| | | | |
|----|------|-----------------|--|
| | | | <p>0 = function data that was configured to be sent to a remote server via UDP</p> <p>2 = function data that was configured to be sent to a remote server via SMS only</p> <p>3 = function data that was configured to be sent to a remote server via TCPAPI only</p> <p>4 = function data that was configured to be save in the buffer</p> <p><param2></p> <p>0: Send to first server defined</p> <p>1: Send to the serial port</p> |
| 49 | 1-2 | 1 | <p>Reset the GPS</p> <p>1: cool boot</p> <p>2: warm boot</p> |
| 50 | 1 | 1 | Reset the communication module |
| 51 | 1 | 1 | Reset the device |
| 52 | 0-16 | 0-1 | <p>Set the time of day function as a current RTC time</p> <p><param1> function index</p> <p><param2></p> <p>16-31 = period</p> <p>0-16 bits = frequency</p> <p>1 = minutes</p> <p>2 = hours</p> <p>4 = days</p> <p>8 = 30 days period</p> |
| 53 | 1-32 | 1 to 2147483647 | <p>Increments counter indicated by parm1 by value of parm2 (for example, if parm1 is 4, parm2 is 100, and counter 4 was 200 prior to this function, counter 4 would be incremented to 300 by this function)</p> <p>It is applicable only for counter type 7</p> |
| 54 | 1-32 | 1 to 2147483647 | <p>Decrements counter indicated by parm1 by value of parm2 (for example, if parm1 is 4, parm2 is 100, and counter 4 was 200 prior to this function, counter 4 would be</p> |

| | | | |
|----|---------|-------------------------|--|
| | | | incremented to 100 by this function) It is applicable only for counter type 7 |
| 55 | 1-32 | 0 | Disable counter indicated by parm1 |
| 56 | 1-32 | 0 | Enable counter indicated by parm1 |
| 57 | 0 to -1 | See Messages type Table | Generate and transmit message to auxiliary serial port based on Parm1 and Parm2 values in ASCII. If messages format is in binary the information will be show in hexadecimal format |
| 58 | 0 – 9 | 0-9 | Add the value of the user variable indicated by parm2 to the user variable indicated by parm1 (for example, if parm1 is 9, parm2 is 2, and user variable 9 was 50 and the user variable 2 was 5 prior to this function, user variable 9 would be incremented to 55 by this function) |

User Variable Index Table

This table is used only with Output Function 30

| Parm2 | System Variable Copied to User Variable |
|--------------|---|
| | (For example, AT\$TTFNT=99,3,128,3,10 would copy value of Input Function 10 (GSM registration status) into User Variable 3). NOTE: All the following system variables are not supported by all devices. Ensure your device supports the system variable before attempting to use it with user variables. |
| -302 | Copies Software version (for example, if version is A02V12, value would be 0x00000212) |
| -301 | Copies product ID (it is an identification of the hardware) |
| -300 | Copies \$ttusrval value |
| -299 to -158 | Reserved |
| -157 | Copies GPS heading from last valid GPS data: \$GPRMC heading (in degrees) times 10: For example: If \$GPRMC reports 084.4 degrees, value will be stored as: |

| | |
|-------------|--|
| | 844 = 0x0000034c |
| -156 | Copies # of GPS satellites from last valid GPS data |
| -155 | Copies GPS velocity from last valid GPS data: \$GPRMC velocity (in knots) times 10. For example: If \$GPRMC reports 022.4 (22.4 knots), value will be stored as: 224 = 0x000000e0 |
| -154 | Copies GPS altitude (in meters) from last valid GPS data |
| -153 | Copies GPS longitude from last valid GPS data: \$GPRMC longitude times 10000 (if 'W' times -1). For example, If \$GPRMC reports 01131.000,E (11 deg 31.000' E), value will be stored as: 11310000 = 0x00ac93b0 If \$GPRMC reports 01131.000,W (11 deg 31.000' W), value will be stored as: -11310000 = 0xff536c50 |
| -152 | Copies GPS latitude from last valid GPS data: \$GPRMC latitude times 10000 (if 'S' times -1). For example, If \$GPRMC reports 4807.038,N (48 deg 07.038' N), value will be stored as: 48070380 = 0x02dd7eec If \$GPRMC reports 4807.038,S (48 deg 07.038' S), value will be stored as: -48070380 = 0xfd228114 |
| -151 | Copies GPS time from last valid GPS data: HHMMSS (HH=\$GPRMC hour, MM=\$GPRMC minute, SS=\$GPRMC second) For example, to convert to hours, minutes, and seconds, use the following: secs = usrvar MOD 100 hours = usrvar DIV 10000 mins = (usrvar MOD 10000) DIV 1000 |
| -150 | Copies GPS date from last valid GPS data: DDMMYY (DD=\$GPRMC day, MM=\$GPRMC month, YY=\$GPRMC year) |
| -149 to -42 | Reserved |
| -41 | Copies GPIO value status where LSB represents GPIO0. For example: 0x00165432 2=binary 0010 (so GPIO1=1; GPIO0, GPIO2, GPIO3=0) 3=binary 0011 (so GPIO4,GPIO5=1; GPIO6,GPIO7=0) |

| | |
|------------|---|
| | <p>4=binary 0100 (so GPIO10=1; GPIO8,GPIO9,GPIO11=0) 5=binary 0101 (so GPIO12,GPIO14=1; GPIO13,GPIO15=0) 6=binary 0110 (so GPIO17,GPIO18=1; GPIO16,GPIO19=0) 1=binary 0001 (so GPIO20=1) Note: GPIO numbers>8 are not supported on the TTXXXX</p> |
| -40 | <p>Copies GPIO direction status where LSB represents GPIO0 (1=input, 0=output). For example: 0x00165432 2=binary 0010 (so GPIO1=input; GPIO0, GPIO2, GPIO3= outputs) 3=binary 0011 (so GPIO4,GPIO5=inputs; GPIO6,GPIO7=outputs) 4=binary 0100 (so GPIO10=input; GPIO8,GPIO9,GPIO11=outputs) 5=binary 0101 (so GPIO12,GPIO14=inputs; GPIO13,GPIO15=outputs) 6=binary 0110 (so GPIO17,GPIO18=inputs; GPIO16,GPIO19=outputs) 1=binary 0001 (so GPIO20=input) Note: GPIO numbers>8 are not supported on the TTXXXX</p> |
| -39 to -35 | Reserved |
| -34 | Copies current RTC time: 0x00HHMMSS where HH = hour (0-23), MM = minute (0-59), SS = second (0-59) |
| -33 | Copies current RTC date: 0x00YYMMDD where YY = last two digits of year (00-99), MM = month (1-12), DD = day of month (1-31) |
| -32 | Copies current count of counter 32 (equivalent to \$TTCNTQRY=32) |
| -31 | Copies current count of counter 31 (equivalent to \$TTCNTQRY=31) |
| -30 | Copies current count of counter 30 (equivalent to \$TTCNTQRY=30) |
| -29 | Copies current count of counter 29 (equivalent to \$TTCNTQRY=29) |
| -28 | Copies current count of counter 28 (equivalent to \$TTCNTQRY=28) |
| -27 | Copies current count of counter 27 (equivalent to \$TTCNTQRY=27) |
| -26 | Copies current count of counter 26 (equivalent to \$TTCNTQRY=26) |
| -25 | Copies current count of counter 25 (equivalent to \$TTCNTQRY=25) |
| -24 | Copies current count of counter 24 (equivalent to \$TTCNTQRY=24) |
| -23 | Copies current count of counter 23 (equivalent to \$TTCNTQRY=23) |
| -22 | Copies current count of counter 22 (equivalent to \$TTCNTQRY=22) |
| -21 | Copies current count of counter 21 (equivalent to \$TTCNTQRY=21) |

| | |
|-----|---|
| -20 | Copies current count of counter 20 (equivalent to \$TTCNTQRY=20) |
| -19 | Copies current count of counter 19 (equivalent to \$TTCNTQRY=19) |
| -18 | Copies current count of counter 18 (equivalent to \$TTCNTQRY=18) |
| -17 | Copies current count of counter 17 (equivalent to \$TTCNTQRY=17) |
| -16 | Copies current count of counter 16 (equivalent to \$TTCNTQRY=16) |
| -15 | Copies current count of counter 15 (equivalent to \$TTCNTQRY=15) |
| -14 | Copies current count of counter 14 (equivalent to \$TTCNTQRY=14) |
| -13 | Copies current count of counter 13 (equivalent to \$TTCNTQRY=13) |
| -12 | Copies current count of counter 12 (equivalent to \$TTCNTQRY=12) |
| -11 | Copies current count of counter 11 (equivalent to \$TTCNTQRY=11) |
| -10 | Copies current count of counter 10 (equivalent to \$TTCNTQRY=10) |
| -9 | Copies current count of counter 9 (equivalent to \$TTCNTQRY=9) |
| -8 | Copies current count of counter 8 (equivalent to \$TTCNTQRY=8) |
| -7 | Copies current count of counter 7 (equivalent to \$TTCNTQRY=7) |
| -6 | Copies current count of counter 6 (equivalent to \$TTCNTQRY=6) |
| -5 | Copies current count of counter 5 (equivalent to \$TTCNTQRY=5) |
| -4 | Copies current count of counter 4 (equivalent to \$TTCNTQRY=4) |
| -3 | Copies current count of counter 3 (equivalent to \$TTCNTQRY=3) |
| -2 | Copies current count of counter 2 (equivalent to \$TTCNTQRY=2) |
| -1 | Copies current count of counter 1 (equivalent to \$TTCNTQRY=1) |
| 0 | Copies value of Input Function 0 (GPIO1). 0 = Low 1 = High |
| 1 | Copies value of Input Function 1 (GPIO2). 0 = Low 1 = High |
| 2 | Copies value of Output Function 3 (GPIO3). 0 = Low 1 = High |
| 3 | Copies value of Input Function 3 (GPIO4). |

| | |
|----|---|
| | 0 = Low 1 = High |
| 4 | Copies value of Output Function 5 (GPIO5). 0 = Low 1 = High |
| 5 | Copies value of Output Function 6 (GPIO6). 0 = Low 1 = High |
| 6 | Copies value of Output Function 7 (GPIO7). 0 = Low 1 = High |
| 7 | Copies value of Output Function 8 (GPIO8). 0 = Low 1 = High |
| 8 | Copies value of Output Function 9 (GPIO9). 0 = Low 1 = High |
| 9 | Reserved |
| 10 | Copies value of Input Function 11 (modem GSM registration). See AT+CREG command description for GSM registration status information. |
| 11 | Copies value of Input Function 12 (modem GPRS registration). See AT%CGREG command for GPRS registration status information. |
| 12 | Copies value of Input Function 13 (Receipt of IP address). 0 = No IP address 1 = Valid IP address obtained |
| 13 | Reserved |
| 14 | Copies value of Input Function 16 (ADC1 status) |
| 15 | Copies value of Input Function 17 (ADC2 status) |
| 16 | Copies value of Input Function 19 (GPS status) 0 = Invalid GPS data 1 = Valid GPS data |

| | |
|----|--|
| 17 | Copies value of Input Function 20 (RTC Alarm Input) |
| 18 | Copies value of Input Function 22 (Invalid GPS data for a period of seconds) |
| 19 | Copies value of Input Function 23 (device staying Idle in one place for a period of seconds) |
| 20 | Copies value of Input Function 49 (Receipt of incoming call with Call Identifier matching one of the numbers configured via the \$TTCALDLST command) |
| 21 | Copies value of Odometer Function 26 (Odometer in meters) |
| 22 | Copies value of Input Function 46 (GPS Antenna status) 0 = unknown 1 = good 2 = open 3 = short |
| 23 | Copies value of Input Function 27 (GPS over speed) 0 = interval has ended 1 = interval has begin |

GPIO Flash Table

| Parm1 | Parm2 |
|--|--|
| Bits 16 – 31 determine the low signal state while bits 0 – 15 determine the high signal state. A value of 0 for bits 16 – 31 indicates the GPIO will remain in low signal state for the same amount of time as the high signal state (50% duty cycle). The high or low states are measured in multiples of ¼ seconds. The toggle count is set by Parm2 | The flashing GPIO function will cause the GPIO output state to toggle at time 0 to the opposite state prior to starting the GPIO output flash function processing. This counts as toggle #1. An even number of toggle count will force a final state which is the same as the initial state. An odd number of toggle count will force the final state to be opposite of the initial GPIO output condition. 0 = toggle forever. |

Message Type Table

| Parm2 | Description |
|-------|-------------|
|-------|-------------|

| | |
|---|--------------------------|
| 0 | Keep alive message |
| 1 | Position report messages |
| 2 | Counter messages |
| 3 | Variables messages |
| 4 | Geofence messages |

Keep Alive Message

The message mask value is obtained as a result of selecting individual bit-fields from the table below.

| Message Mask | Description |
|--------------|---|
| Bit 0 | 1 = send all data generated as a result of this table in Binary format 0 = send all data generated as a result of this table in ASCII format |
| Bit 1 | 1 = add parm1 data to UDP message (4 – bytes in Binary format, 11 – bytes of data in ASCII format) 0 = do not add parm1 data to outbound UDP message |
| Bit 2 | 1 = add \$TTDEVID value (22 bytes in binary; 22 – bytes plus 1 byte of space character in ASCII) 0 = do not add \$TTDEVID value (In ASCII format all of the fields are separated by a space character except this field which has one additional space character) |
| Bits 3-26 | Reserved |
| Bit 27 | 1 = Add Keep Alive report mask (4 – bytes in Binary format, 4 – bytes in ASCII format) 0 = do not add Keep Alive mask |
| Bit 28 | 1 = Add sequence number message (2 – bytes in Binary format, 4 – bytes of data in ASCII format) 0 = Do not add sequence number |
| Bits 29-31 | Reserved |

Position Report Messages

The message mask value is obtained as a result of selecting individual bit-fields from the table below.

| Message Mask | Description |
|--------------|---|
| Bit 0 | 1 = send all data generated as a result of this table in Binary format 0 = send all data generated as a result of this table in ASCII format |
| Bit 1 | 1 = add parm1 data to UDP message (4 – bytes in Binary format, 11 – bytes of data in ASCII format) 0 = do not add parm1 data to outbound UDP message |
| Bit 2 | 1 = add \$TTDEVID value (22 bytes in binary; 22 – bytes plus 1 byte of space character in ASCII) 0 = do not add \$TTDEVID value (In ASCII format all of the fields are separated by a space character except this field which has one additional space character) |
| Bit 3 | 1 = add the I/O status (GPIO direction) in ASCII-DEC format (2 – bytes in Binary format, 6 – bytes in ASCII format) 0 = do not add GPIO direction and data value. |
| Bit 4 | 1 = add Analog input 1 to UDP message (2 – bytes in Binary format, 5 – bytes of data in ASCII format) 0 = do not add Analog input 1 data to outbound UDP message |
| Bit 5 | 1 = add Analog input 2 to UDP message (2 – bytes in Binary format, 5 – bytes of data in ASCII format) 0 = do not add Analog input 1 data to outbound UDP message |
| Bit 6 | 1 = Message is stored in non-volatile memory until it can be sent, regardless of network status. 0 = Code checks network status before storing message in non-volatile memory. If it appears that the message can be sent out immediately (network status is clear and message queue has few or no messages pending), the message is stored in the non-volatile message queue until it can be sent. Otherwise, the message is deleted. |
| Bit 7 | 1 = add input <function category> number (1 – byte in binary format, 3 – bytes in ASCII format) 0 = do not add input <function category> number |

| | |
|--------|---|
| Bit 8 | 1 = add GPS data (3 – bytes of Date information in Binary format or up to 80 – bytes of \$GPGGA NMEA message if Bit-0 is set to 0) 0 = do not add this particular field of GPS data |
| Bit 9 | 1 = add 1-byte of STATUS information in Binary format or 0 bytes if Bit-0 is set to 0 0 = do not add this particular field of GPS data |
| Bit 10 | 1 = add GPS data (4 – bytes of Latitude information in Binary format or up to 80 – bytes of \$GPGSA NMEA message if Bit-0 is set to 0) 0 = do not add this particular field of GPS data |
| Bit 11 | 1 = add GPS data (4 – bytes of Longitude information in Binary format or up to two 80 – bytes of \$GPGSV NMEA message if Bit-0 is set to 0) 0 = do not add this particular field of GPS data |
| Bit 12 | 1 = add GPS data (2 – bytes of Velocity information in Binary format or up to 80 – bytes of \$GPRMC NMEA message if Bit-0 is set to 0) 0 = do not add this particular field of GPS data |
| Bit 13 | 1 = add 2-bytes of HEADING information in Binary format or 0 bytes if Bit-0 is set to 0 0 = do not add this particular field of GPS data |
| Bit 14 | 1 = add GPS data (3 – bytes of Time information in Binary format or 0 bytes if Bit-0 is set to 0) 0 = do not add this particular field of GPS data |
| Bit 15 | 1 = add GPS data (3 – bytes of Altitude information in Binary format or 0 bytes if Bit-0 is set to 0) 0 = do not add this particular field of GPS data |
| Bit 16 | 1 = add GPS data (1 – byte of Number Of Satellites In View information in Binary format or 0 bytes if Bit-0 is set to 0) 0 = do not add this particular field of GPS data |
| Bit 17 | 1 = add internal battery level percentage (2 - bytes of in Binary format or 3 - bytes if Bit-0 is set to 0) 0 = do not add this particular field |
| Bit 18 | 1 = send this OTA message via SMS when GPRS services is not available 0 = send this OTA message via GPRS only |
| Bit 19 | 1 = send Last Valid GPS data if current data is invalid 0 = send current GPS data – valid or invalid |

| | |
|--------|--|
| Bit 20 | <p>1 = add Trip Odometer reading (4 - bytes of Trip Odometer information in Binary format or 11 - bytes if Bit-0 is set to 0)</p> <p>0 = do not add this particular field of GPS data</p> <p>NOTE: The Trip Odometer is associated with the AT\$TTTODOM command.</p> |
| Bit 21 | <p>1 = add Odometer reading (4 - bytes of Trip Odometer information in Binary format or 11 - bytes if Bit-0 is set to 0)</p> <p>0 = do not add this particular field of GPS data</p> <p>NOTE: The Trip Odometer is associated with the AT\$TTODOM command.</p> |
| Bit 22 | <p>1 = add RTC time (6 – bytes of RTC time in Binary format or 14 – bytes if Bit-0 is set to 0)</p> <p>0 = do not add RTC time with GPS data</p> |
| Bit 23 | <p>1 = Replace/append device id field with 10-byte device id (including one leading and one ending space character) if bit-0 is set to 0. Replace/append it with 8-bytes long device id value if bit-0 is set to 1 (no leading or ending space characters in binary mode.) (NOTE: bit-22 setting overrides bit-2 setting)</p> <p>0 = Sent the device id as defined by Bit-2</p> |
| Bit 24 | <p>1 = add main power voltage (2 - bytes of in Binary format or 5 - bytes if Bit-0 is set to 0)</p> <p>0 = do not add this particular field</p> |
| Bit 25 | <p>1 = add GPS overspeed data (6 – bytes of Odometer information in Binary format or 6 to 18 bytes if Bit-0 is set to 0).</p> <p>Binary format: xxyyzz:</p> <ul style="list-style-type: none"> xx is speed specified by AT\$TTOVSDEF (unit: knots); yy is the maximum speed incurred during the interval (unit: knots, 1/10 knot accuracy); zz is the interval duration (unit: seconds); <p>ASCII format:</p> <p>“ x y z”: space delineated, length of each field varies with its value</p> <p>0 = do not add this particular field of GPS data</p> |
| Bit 26 | <p>1 = Add cell information as follows:</p> <p>If Binary format (Bit0=1) is selected, please refer to the “Bit 25 Binary Format” table in section (Bit 26 Binary Format Table)</p> <p>If ASCII format (Bit0=0) is selected please refer to the “Bit 25 ASCII Format” table in section (Bit 26 ASCII Format Table)</p> |

| | |
|--------------|---|
| | 0 = Do not add cell information |
| Bit27 | 1 = Add Position report mask (4 – bytes in Binary format, 4 – bytes in ASCII format) 0 = do not add Position report mask |
| Bit 28 | 1 = Add sequence number message (2 – bytes in Binary format, 4 – bytes of data in ASCII format) 0 = Do not add sequence number |
| Bits 29 - 31 | Reserved |

Bit 26 Binary Format Table

| Bit 25 Binary Format Table | |
|------------------------------|---|
| MCCMNC | Mobile Country Code Mobile Network Code (4 bytes) |
| Serving Cell LAC | Location Area Code (2 bytes) |
| Serving Cell CellID | Cell Tower Identification (2 bytes) |
| Serving Cell ARFCN | Absolute radio frequency channel number (2 bytes) |
| Serving Cell signal strength | Cell signal strength (1 byte) |
| Serving Cell timing advance | Cell timing advance (1 byte) |
| Neighbor Cell 0 LAC | LAC for Neighbor Cell Tower 0 (2 bytes) |
| Neighbor Cell 0 CellID | CellID for Neighbor Cell Tower 0 (2 bytes) |
| Neighbor Cell 0 ARFCN | ARFCN for Neighbor Cell Tower 0 (2 bytes) |
| Neighbor Cell 0 rxlev | Signal quality for Neighbor Cell Tower 0 (1 byte) |
| Neighbor Cell 1 LAC | LAC for Neighbor Cellular tower 1 (2 bytes) |
| Neighbor Cell 1 CellID | CellID for Neighbor Cell Tower 1 (2 bytes) |
| Neighbor Cell 1 ARFCN | ARFCN for Neighbor Cell Tower 1 (2 bytes) |
| Neighbor Cell 1 rxlev | Signal quality for Neighbor Cell Tower 1 (1 byte) |
| Neighbor Cell 2 LAC | LAC for Neighbor Cellular tower 2 (2 bytes) |
| Neighbor Cell 2 CellID | CellID for Neighbor Cell Tower 2 (2 bytes) |
| Neighbor Cell 2 ARFCN | ARFCN for Neighbor Cell Tower 2 (2 bytes) |
| Neighbor Cell 2 rxlev | Signal quality for Neighbor Cell Tower 2 (1 byte) |

| | |
|------------------------|--|
| Neighbor Cell 3 LAC | LAC for Neighbor Cellular tower 3 (2 bytes) |
| Neighbor Cell 3 CellID | CellID for Neighbor Cell Tower 3 (2 bytes) |
| Neighbor Cell 3 ARFCN | ARFCN for Neighbor Cell Tower 3 (2 bytes) |
| Neighbor Cell 3 rxlev | Signal quality for Neighbor Cell Tower 3 (1 byte) |
| Neighbor Cell 4 LAC | LAC for Neighbor Cellular tower 4 (2 bytes) |
| Neighbor Cell 4 CellID | CellID for Neighbor Cell Tower 4 (2 bytes) |
| Neighbor Cell 4 ARFCN | ARFCN for Neighbor Cell Tower 4 (2 bytes) |
| Neighbor Cell 4 rxlev | Signal quality for Neighbor Cell Tower 4 (1 byte) |
| Neighbor Cell 5 LAC | LAC for Neighbor Cellular tower 5 (2 bytes) |
| Neighbor Cell 5 CellID | CellID for Neighbor Cell Tower 5 (2 bytes) |
| Neighbor Cell 5 ARFCN | ARFCN for Neighbor Cell Tower 5 (2 bytes) |
| Neighbor Cell 5 rxlev | Signal quality for Neighbor Cell Tower 5 (1 byte) |

Bit 26 ASCII Format Table

| Bit 26 ASCII Format Table | |
|--|---|
| If ASCII format (Bit0=0) is selected: Variable length string is appended to message with semicolons separating cells (Serving Cells and Neighbor Cells) and commas separating the fields within a cell as follows: mcc,mnc,sc_lac,sc_cell_id,sc_arfcn,sc_rxlev,sc_tav; lac_nc0,cell_id_nc0,arfcn_nc0,rxlev_nc0; lac_nc1,cell_id_nc1,arfcn_nc1,rxlev_nc1; lac_nc2,cell_id_nc2,arfcn_nc2,rxlev_nc2; lac_nc3,cell_id_nc3,arfcn_nc3,rxlev_nc3; lac_nc4,cell_id_nc4,arfcn_nc4,rxlev_nc4; lac_nc5,cell_id_nc5,arfcn_nc5,rxlev_nc5 | |
| mcc | Mobile Country Code |
| mnc | Mobile Network Code |
| sc_lac | Location Area Code |
| sc_cell_id | Cell Tower Identification |
| sc_arfcn | Absolute radio frequency channel number |

| | |
|-------------|--|
| sc_rxlev | Cell signal strength |
| sc_tav | Cell timing advance |
| lac_nc0 | LAC for Neighbor Cell Tower 0 |
| cell_id_nc0 | CellID for Neighbor Cell Tower 0 |
| arfcn_nc0 | ARFCN for Neighbor Cell Tower 0 |
| rxlev_nc0 | Signal quality for Neighbor Cell Tower 0 |
| lac_nc1 | LAC for Neighbor Cell Tower 1 |
| cell_id_nc1 | CellID for Neighbor Cell Tower 1 |
| arfcn_nc1 | ARFCN for Neighbor Cell Tower 1 |
| rxlev_nc1 | Signal quality for Neighbor Cell Tower 1 |
| lac_nc2 | LAC for Neighbor Cell Tower 2 |
| cell_id_nc2 | CellID for Neighbor Cell Tower 2 |
| arfcn_nc2 | ARFCN for Neighbor Cell Tower 2 |
| rxlev_nc2 | Signal quality for Neighbor Cell Tower 2 |
| lac_nc3 | LAC for Neighbor Cell Tower 3 |
| cell_id_nc3 | CellID for Neighbor Cell Tower 3 |
| arfcn_nc3 | ARFCN for Neighbor Cell Tower 3 |
| rxlev_nc3 | Signal quality for Neighbor Cell Tower 3 |
| lac_nc4 | LAC for Neighbor Cell Tower 4 |
| cell_id_nc4 | CellID for Neighbor Cell Tower 4 |
| arfcn_nc4 | ARFCN for Neighbor Cell Tower 4 |
| rxlev_nc4 | Signal quality for Neighbor Cell Tower 4 |
| lac_nc5 | LAC for Neighbor Cell Tower 5 |
| cell_id_nc5 | CellID for Neighbor Cell Tower 5 |
| arfcn_nc5 | ARFCN for Neighbor Cell Tower 5 |
| rxlev_nc5 | Signal quality for Neighbor Cell Tower 5 |

Counter messages

The Parm2 value is obtained as a result of selecting individual bit-fields from the table below.

| Bit-Field Table 2 – (1,0) | |
|---------------------------|---|
| Parm2 | Description |
| Bit 0 | 1 = send all data generated as a result of this table in Binary format 0 = send all data generated as a result of this table in ASCII format |
| Bit 1 | 1 = Add parm1 data to message (4 bytes in binary format, 11 bytes of data in ASCII format) 0 = do not add parm1 data to outbound message |
| Bit 2 | 1 = add \$TTDEVID value (22 bytes in binary; 22 – bytes plus 1 byte of space character in ASCII) 0 = do not add \$TTDEVID value (In ASCII format all of the fields are separated by a space character except this field which has one additional space character) |
| Bit 3 | 1 = add active counters to message, first value is the number of the counter included un the message. In ASCII format the values are separated by comma (,); in binary format, 10 bytes per each counter Binary format: nlllllvvv where n: counter index, t: counter type, llll: limit, vvv: current value ASCII format: n,t,l,v where n: counter index, t: counter type, l: limit, v: current value 0 = do not add user variables |
| Bit 4 | 1 = add Analog input 1 to UDP message (2 – bytes in Binary format, 5 – bytes of data in ASCII format) 0 = do not add Analog input 1 data to outbound UDP message |
| Bit 5 | 1 = add Analog input 2 to UDP message (2 – bytes in Binary format, 5 – bytes of data in ASCII format) 0 = do not add Analog input 1 data to outbound UDP message |
| Bit 6: | 1 =Message is stored in non-volatile memory until it can be sent, regardless of network status. 0 = Check network status before storing message in non-volatile memory. If it appears that the message can be sent out immediately (network status is clear and message queue has few or no messages pending), the message is stored in the non-volatile message queue until it can be sent. Otherwise, |

| | |
|--------|---|
| | the message is deleted. |
| Bit 7 | 1 = add input <function category> number (1 byte in binary format, 3 bytes in ASCII format) 0 = do not add input <function category> number |
| Bit 8 | 1 = add GPS data (3 – bytes of Date information in Binary format or up to 80 – bytes of \$GPGGA NMEA message if Bit-0 is set to 0) 0 = do not add this particular field of GPS data |
| Bit 9 | 1 = add 1-byte of STATUS information in Binary format or 0 bytes if Bit-0 is set to 0 0 = do not add this particular field of GPS data |
| Bit 10 | 1 = add GPS data (4 – bytes of Latitude information in Binary format or up to 80 – bytes of \$GPGSA NMEA message if Bit-0 is set to 0) 0 = do not add this particular field of GPS data |
| Bit 11 | 1 = add GPS data (4 – bytes of Longitude information in Binary format or up to two 80 – bytes of \$GPGSV NMEA message if Bit-0 is set to 0) 0 = do not add this particular field of GPS data |
| Bit 12 | 1 = add GPS data (2 – bytes of Velocity information in Binary format or up to 80 – bytes of \$GPRMC NMEA message if Bit-0 is set to 0) 0 = do not add this particular field of GPS data |
| Bit 13 | 1 = add 2-bytes of HEADING information in Binary format or 0 bytes if Bit-0 is set to 0 0 = do not add this particular field of GPS data |
| Bit 14 | 1 = add GPS data (3 – bytes of Time information in Binary format or 0 bytes if Bit-0 is set to 0) 0 = do not add this particular field of GPS data |
| Bit 15 | 1 = add GPS data (3 – bytes of Altitude information in Binary format or 0 bytes if Bit-0 is set to 0) 0 = do not add this particular field of GPS data |
| Bit 16 | 1 = add GPS data (1 – byte of Number Of Satellites In View information in Binary format or 0 bytes if Bit-0 is set to 0) 0 = do not add this particular field of GPS data |
| Bit 17 | 1 = add internal battery level percentage (2 - bytes of in Binary format or 3 - bytes if Bit-0 is set to 0) 0 = do not add this particular field |

| | |
|--------|--|
| Bit 18 | <p>1 = send this OTA message via SMS when GPRS services is not available 0 = send this OTA message via GPRS only</p> |
| Bit 19 | <p>1 = send Last Valid GPS data if current data is invalid 0 = send current GPS data – valid or invalid</p> |
| Bit 20 | <p>1 = add Trip Odometer reading (4 - bytes of Trip Odometer information in Binary format or 11 - bytes if Bit-0 is set to 0) 0 = do not add this particular field of GPS data NOTE: The Trip Odometer is associated with the AT\$TTTODOM command.</p> |
| Bit 21 | <p>1 = add Odometer reading (4 - bytes of Trip Odometer information in Binary format or 11 - bytes if Bit-0 is set to 0) 0 = do not add this particular field of GPS data NOTE: The Trip Odometer is associated with the AT\$TTODOM command.</p> |
| Bit 22 | <p>1 = add RTC time (6 bytes of RTC time in binary format or 14 bytes in ASCII format) 0 = do not add RTC time</p> |
| Bit 23 | <p>1 = Replace/append device id field with 10-byte device id (including one leading and one ending space character) in ASCII format. Replace/append device id with 8 bytes long device id value in binary format (no leading or ending space characters in binary format). (NOTE: Bit 22 setting overrides Bit 2 setting) 0 = send the device id as defined by Bit 2</p> |
| Bit 24 | <p>1 = add main power voltage (2 - bytes of in Binary format or 5 - bytes if Bit-0 is set to 0) 0 = do not add this particular field</p> |
| Bit 25 | <p>1 = add GPS overspeed data (6 – bytes of Odometer information in Binary format or 6 to 18 bytes if Bit-0 is set to 0). Binary format: xxyyzz: xx is speed specified by AT\$TTOVSDEF (unit: knots); yy is the maximum speed incurred during the interval (unit: knots, 1/10 knot accuracy); zz is the interval duration (unit: seconds); ASCII format: " x y z": space delineated, length of each field varies with its value 0 = do not add this particular field of GPS data</p> |
| Bit 26 | <p>1 = Add cell information as follows:</p> |

| | |
|------------|--|
| | <p>If Binary format (Bit0=1) is selected, please refer to the “Bit 25 Binary Format” table in section (Bit 26 Binary Format Table)</p> <p>If ASCII format (Bit0=0) is selected please refer to the “Bit 25 ASCII Format” table in section (Bit 26 ASCII Format Table)</p> <p>0 = Do not add cell information</p> |
| Bits 27 | <p>1 = Add Position report mask (4 – bytes in Binary format, 4 – bytes of data in ASCII format)</p> <p>0 = do not add Position report mask</p> |
| Bits 28 | <p>1 = Add sequence number message (2 – bytes in Binary format, 4 – bytes of data in ASCII format)</p> <p>0 = Do not add sequence number</p> |
| Bits 29-31 | Reserved |

Variable Message

The Parm2 value is obtained as a result of selecting individual bit-fields from the table below.

| Bit-Field Table 2 – (1,0) | |
|---------------------------|--|
| Parm2 | Description |
| Bit 0 | <p>1 = send all data generated as a result of this table in Binary format</p> <p>0 = send all data generated as a result of this table in ASCII format</p> |
| Bit 1 | <p>1 = Add parm1 data to message (4 bytes in binary format, 11 bytes of data in ASCII format)</p> <p>0 = do not add parm1 data to outbound message</p> |
| Bit 2 | <p>1 = add \$TTDEVID value (22 bytes in binary; 22 – bytes plus 1 byte of space character in ASCII)</p> <p>0 = do not add \$TTDEVID value</p> <p>(In ASCII format all of the fields are separated by a space character except this field which has one additional space character)</p> |
| Bit 3 | <p>1 = add user variables 0-9 to message, starting with user variable 0 (4 bytes per user variable in binary format, 8 hex bytes per user variable in ASCII format)</p> <p>0 = do not add user variables</p> |
| Bit 4 | <p>1 = add Analog input 1 to UDP message (2 – bytes in Binary format, 5 – bytes of data in ASCII format)</p> |

| | |
|--------|--|
| | 0 = do not add Analog input 1 data to outbound UDP message |
| Bit 5 | 1 = add Analog input 2 to UDP message (2 – bytes in Binary format, 5 – bytes of data in ASCII format) 0 = do not add Analog input 1 data to outbound UDP message |
| Bit 6: | 1 =Message is stored in non-volatile memory until it can be sent, regardless of network status. 0 = Check network status before storing message in non-volatile memory. If it appears that the message can be sent out immediately (network status is clear and message queue has few or no messages pending), the message is stored in the non-volatile message queue until it can be sent. Otherwise, the message is deleted. |
| Bit 7 | 1 = add input <function category> number (1 byte in binary format, 3 bytes in ASCII format) 0 = do not add input <function category> number |
| Bit 8 | 1 = add GPS data (3 – bytes of Date information in Binary format or up to 80 – bytes of \$GPGGA NMEA message if Bit-0 is set to 0) 0 = do not add this particular field of GPS data |
| Bit 9 | 1 = add 1-byte of STATUS information in Binary format or 0 bytes if Bit-0 is set to 0 0 = do not add this particular field of GPS data |
| Bit 10 | 1 = add GPS data (4 – bytes of Latitude information in Binary format or up to 80 – bytes of \$GPGSA NMEA message if Bit-0 is set to 0) 0 = do not add this particular field of GPS data |
| Bit 11 | 1 = add GPS data (4 – bytes of Longitude information in Binary format or up to two 80 – bytes of \$GPGSV NMEA message if Bit-0 is set to 0) 0 = do not add this particular field of GPS data |
| Bit 12 | 1 = add GPS data (2 – bytes of Velocity information in Binary format or up to 80 – bytes of \$GPRMC NMEA message if Bit-0 is set to 0) 0 = do not add this particular field of GPS data |
| Bit 13 | 1 = add 2-bytes of HEADING information in Binary format or 0 bytes if Bit-0 is set to 0 0 = do not add this particular field of GPS data |
| Bit 14 | 1 = add GPS data (3 – bytes of Time information in Binary format or 0 bytes if Bit-0 is set to 0) 0 = do not add this particular field of GPS data |

| | |
|--------|---|
| Bit 15 | 1 = add GPS data (3 – bytes of Altitude information in Binary format or 0 bytes if Bit-0 is set to 0) 0 = do not add this particular field of GPS data |
| Bit 16 | 1 = add GPS data (1 – byte of Number Of Satellites In View information in Binary format or 0 bytes if Bit-0 is set to 0) 0 = do not add this particular field of GPS data |
| Bit 17 | 1 = add internal battery level percentage (2 - bytes of in Binary format or 3 - bytes if Bit-0 is set to 0) 0 = do not add this particular field |
| Bit 18 | 1 = send this OTA message via SMS when GPRS services is not available 0 = send this OTA message via GPRS only |
| Bit 19 | 1 = send Last Valid GPS data if current data is invalid 0 = send current GPS data – valid or invalid |
| Bit 20 | 1 = add Trip Odometer reading (4 - bytes of Trip Odometer information in Binary format or 11 - bytes if Bit-0 is set to 0) 0 = do not add this particular field of GPS data NOTE: The Trip Odometer is associated with the AT\$TTTODOM command. |
| Bit 21 | 1 = add Odometer reading (4 - bytes of Trip Odometer information in Binary format or 11 - bytes if Bit-0 is set to 0) 0 = do not add this particular field of GPS data NOTE: The Trip Odometer is associated with the AT\$TTODOM command. |
| Bit 22 | 1 = add RTC time (6 bytes of RTC time in binary format or 14 bytes in ASCII format) 0 = do not add RTC time |
| Bit 23 | 1 = Replace/append device id field with 10-byte device id (including one leading and one ending space character) in ASCII format. Replace/append device id with 8 bytes long device id value in binary format (no leading or ending space characters in binary format). (NOTE: Bit 22 setting overrides Bit 2 setting) 0 = send the device id as defined by Bit 2 |
| Bit 24 | 1 = add main power voltage (2 - bytes of in Binary format or 5 - bytes if Bit-0 is set to 0) 0 = do not add this particular field |
| Bit 25 | 1 = add GPS overspeed data (6 – bytes of Odometer information in Binary format or 6 to 18 bytes if Bit-0 is set to 0). |

| | |
|------------|--|
| | <p>Binary format: xyyzz:</p> <p>xx is speed specified by AT\$TTOVSDEF (unit: knots);</p> <p>yy is the maximum speed incurred during the interval (unit: knots, 1/10 knot accuracy);</p> <p>zz is the interval duration (unit: seconds);</p> <p>ASCII format:</p> <p>“ x y z”: space delineated, length of each field varies with its value</p> <p>0 = do not add this particular field of GPS data</p> |
| Bit 26 | <p>1 = Add cell information as follows:</p> <p>If Binary format (Bit0=1) is selected, please refer to the “Bit 25 Binary Format” table in section (Bit 26 Binary Format Table)</p> <p>If ASCII format (Bit0=0) is selected please refer to the “Bit 25 ASCII Format” table in section (Bit 26 ASCII Format Table)</p> <p>0 = Do not add cell information</p> |
| Bits 27 | <p>1 = Add Position report mask (4 – bytes in Binary format, 4 – bytes of data in ASCII format)</p> <p>0 = do not add Position report mask</p> |
| Bits 28 | <p>1 = Add sequence number message (2 – bytes in Binary format, 4 – bytes of data in ASCII format)</p> <p>0 = Do not add sequence number</p> |
| Bits 29-31 | Reserved |

Geofence Message

The Parm2 value is obtained as a result of selecting individual bit-fields from the table below.

| Bit-Field Table 3 – (1,1) | |
|---------------------------|--|
| Parm2 | Description |
| Bit 0 | <p>1 = send all data generated as a result of this table in Binary format</p> <p>0 = send all data generated as a result of this table in ASCII format</p> |
| Bit 1 | <p>1 = Add parm1 data to message (4 bytes in binary format, 11 bytes of data in ASCII format)</p> <p>0 = do not add parm1 data to outbound message</p> |
| Bit 2 | 1 = add \$TTDEVID value (22 bytes in binary; 22 – bytes plus 1 byte of space) |

| | |
|---------|--|
| | <p>character in ASCII</p> <p>0 = do not add \$TTDEVID value</p> <p>(In ASCII format all of the fields are separated by a space character except this field which has one additional space character)</p> |
| Bits 3 | <p>1 = add \$IOCFG and \$IOGPA (GPIO direction and data) in ASCII-HEX format (2 – bytes in Binary format, 6 – bytes in ASCII format)</p> <p>0 = do not add GPIO direction and data value.</p> |
| Bit 4 | <p>1 = add the information related with last geo-fence triggered (2 - bytes in Binary format , 4 bytes in ASCII format,</p> <p>Binary format: gi where g: Geo fence <index>, i: position of the geofence in the geofence definition;</p> <p>ASCII format: ggii where where g: Geo fence <index>, i: position of the geofence in the geofence definition)</p> <p>0 = do not add the geofence information</p> |
| Bit 5 | <p>1 = add Analog input 2 to UDP message (2 – bytes in Binary format, 5 – bytes of data in ASCII format)</p> <p>0 = do not add Analog input 1 data to outbound UDP message</p> |
| Bit 6: | <p>1 = Message is stored in non-volatile memory until it can be sent, regardless of network status.</p> <p>0 = Check network status before storing message in non-volatile memory. If it appears that the message can be sent out immediately (network status is clear and message queue has few or no messages pending), the message is stored in the non-volatile message queue until it can be sent. Otherwise, the message is deleted.</p> |
| Bit 7 | <p>1 = add input <function category> number (1 byte in binary format, 3 bytes in ASCII format)</p> <p>0 = do not add input <function category> number</p> |
| Bits 8 | <p>1 = add GPS data (3 – bytes of Date information in Binary format or up to 80 – bytes of \$GPGGA NMEA message if Bit-0 is set to 0)</p> <p>0 = do not add this particular field of GPS data</p> |
| Bits 9 | <p>1 = add 1-byte of STATUS information in Binary format or 0 bytes if Bit-0 is set to 0</p> <p>0 = do not add this particular field of GPS data</p> |
| Bits 10 | <p>1 = add GPS data (4 – bytes of Latitude information in Binary format or up to 80 – bytes of \$GPGSA NMEA message if Bit-0 is set to 0)</p> <p>0 = do not add this particular field of GPS data</p> |

| | |
|---------|--|
| Bits 11 | <p>1 = add GPS data (4 – bytes of Longitude information in Binary format or up to two 80 – bytes of \$GPGSV NMEA message if Bit-0 is set to 0)</p> <p>0 = do not add this particular field of GPS data</p> |
| Bits 12 | <p>1 = add GPS data (2 – bytes of Velocity information in Binary format or up to 80 – bytes of \$GPRMC NMEA message if Bit-0 is set to 0)</p> <p>0 = do not add this particular field of GPS data</p> |
| Bits 13 | <p>1 = add 2-bytes of HEADING information in Binary format or 0 bytes if Bit-0 is set to 0</p> <p>0 = do not add this particular field of GPS data</p> |
| Bits 14 | <p>1 = add GPS data (3 – bytes of Time information in Binary format or 0 bytes if Bit-0 is set to 0)</p> <p>0 = do not add this particular field of GPS data</p> |
| Bits 15 | <p>1 = add GPS data (3 – bytes of Altitude information in Binary format or 0 bytes if Bit-0 is set to 0)</p> <p>0 = do not add this particular field of GPS data</p> |
| Bits 16 | <p>1 = add GPS data (1 – byte of Number Of Satellites In View information in Binary format or 0 bytes if Bit-0 is set to 0)</p> <p>0 = do not add this particular field of GPS data</p> |
| Bits 17 | <p>1 = add battery level percentage (2 - bytes of in Binary format or 3 - bytes if Bit-0 is set to 0)</p> <p>0 = do not add this particular field</p> |
| Bit 18 | <p>1 = send this OTA message via SMS when GPRS services is not available</p> <p>0 = send this OTA message via GPRS only</p> |
| Bits 19 | <p>1 = send Last Valid GPS data if current data is invalid</p> <p>0 = send current GPS data – valid or invalid</p> |
| Bits 20 | <p>1 = add Trip Odometer reading (4 - bytes of Trip Odometer information in Binary format or 11 - bytes if Bit-0 is set to 0)</p> <p>0 = do not add this particular field of GPS data</p> <p>NOTE: The Trip Odometer is associated with the AT\$TTTODOM command.</p> |
| Bits 21 | <p>1 = add Odometer reading (4 - bytes of Trip Odometer information in Binary format or 11 - bytes if Bit-0 is set to 0)</p> <p>0 = do not add this particular field of GPS data</p> <p>NOTE: The Trip Odometer is associated with the AT\$TTODOM command.</p> |
| Bit 22 | <p>1 = add RTC time (6 bytes of RTC time in binary format or 14 bytes in ASCII</p> |

| | |
|------------|---|
| | <p>format)</p> <p>0 = do not add RTC time</p> |
| Bit 23 | <p>1 = Replace/append device id field with 10-byte device id (including one leading and one ending space character) in ASCII format. Replace/append device id with 8 bytes long device id value in binary format (no leading or ending space characters in binary format).</p> <p>(NOTE: Bit 22 setting overrides Bit 2 setting)</p> <p>0 = send the device id as defined by Bit 2</p> |
| Bit 24 | <p>1 = add main power voltage (2 - bytes of in Binary format or 5 - bytes if Bit-0 is set to 0)</p> <p>0 = do not add this particular field</p> |
| Bit 25 | <p>1 = add GPS overspeed data (6 – bytes of Odometer information in Binary format or 6 to 18 bytes if Bit-0 is set to 0).</p> <p>Binary format: xxyyzz:</p> <p>xx is speed specified by AT\$TTOVSDEF (unit: knots);</p> <p>yy is the maximum speed incurred during the interval (unit: knots, 1/10 knot accuracy);</p> <p>zz is the interval duration (unit: seconds);</p> <p>ASCII format:</p> <p>“ x y z”: space delineated, length of each field varies with its value</p> <p>0 = do not add this particular field of GPS data</p> |
| Bit 26 | <p>1 = Add cell information as follows:</p> <p>If Binary format (Bit0=1) is selected, please refer to the “Bit 25 Binary Format” table in section (Bit 26 Binary Format Table)</p> <p>If ASCII format (Bit0=0) is selected please refer to the “Bit 25 ASCII Format” table in section (Bit 26 ASCII Format Table)</p> <p>0 = Do not add cell information</p> |
| Bits 27 | <p>1 = Add Position report mask (4 – bytes in Binary format, 4 – bytes of data in ASCII format)</p> <p>0 = do not add Position report mask</p> |
| Bits 28 | <p>1 = Add sequence number message (2 – bytes in Binary format, 4 – bytes of data in ASCII format)</p> <p>0 = Do not add sequence number</p> |
| Bits 29-31 | Reserved |

Appendix A – Result Codes

Result Codes

| Modem Verbose Response | Modem Terse Response | Definition |
|------------------------|----------------------|-------------------------------------|
| OK | 0 | command successful completed; ready |
| CONNECT | 1 | entering data transfer state |
| RING | 2 | Ring indication detected |
| NO CARRIER | 3 | connection terminated |
| ERROR | 4 | Command abnormally completed, ready |
| NO DIALTONE | 6 | Dial tone not found |
| BUSY | 7 | Busy signal detected |
| NO ANSWER | 8 | connection completion timeout |

Unsolicited Result Codes

| Result Code | Definition | AT Command |
|--|--|------------|
| +CCCM: <ccm> | Current call meter value | AT+CACM=1 |
| +CCWA: <number>,<type> ,<class> [,<alpha>] | Call Waiting Status | AT+CCWA=1 |
| +CLAV: <code> | ME Language Change | AT+CLAE=1 |
| +CLIP: <number>,<type>[,<subaddr> ,<satype>[,<alpha>]] | Calling Line Identification Presentation | AT+CLIP=1 |
| +CME ERROR: <err> | ME Error Result Code | AT+CMEE=x |
| +COLP: <number>,<type>[,<subaddr> ,<satype>[,<alpha>]] | Connected Line Identification Presentation | AT+COLP=1 |
| +CR: <type> | Service Reporting Control | AT+CR=1 |
| +CREG: <stat>[,<lac>,<ci>] | Registration status indication | AT+CREG=1 |

| | | |
|--|------------------------------------|-------------|
| +CRING: <type> | Incoming Call Indication | AT+CRC=1 |
| +CSSI: <code1>[,<index>] | Supplementary Services Result Code | AT+CSSN=1,1 |
| +CSSU: <code2>[,<index>[,<number>, <type>[,<subaddr>,<satype>]]] | Supplementary Services Result Code | AT+CSSN=1,1 |
| +CUSD: <m>[,<str>,<dcs>] | Indication of Incoming USSD String | AT+CUSD=1 |
| +CGREG: <stat>[,<lac>,<ci>] | GPRS Registration Status | AT+CGREG=1 |

SMS Unsolicited Result Codes

| Result Code | Definition | AT Command |
|---|---|------------------------------------|
| +CMTI: <mem>,<index> | Indication of new short message | AT+CNMI=1,1 |
| +CMT: <length><CR><LF><pdu> | Short Message output Directly to TE (PDU mode) | AT+CNMI=1,2 |
| +CBM: <sn>, <mid>,<dcs>,<page>,<pages> <CR><LF><data> | Incoming Cell Broadcast Message routed directly to TE | AT+CNMI=1,0,2 |
| +CDS: <length><CR><LF><pdu> | SMS status report routed directly to the TE | AT+CNMI=1,0,0, 1 AT+CSMP=49, |

Appendix B – Error Codes

CME Error Codes

Final result code **+CME ERROR: <err>** indicates an error related to mobile equipment or network. The operation is similar to **ERROR** result code. None of the following commands in the same command line is executed. Neither **ERROR** nor **OK** result code shall be returned.

<err> values are mostly used by common messaging commands. The following table lists most of general and GRPS related **ERROR** Codes. For some GSM protocol failure cause described in GSM specifications, the corresponding **ERROR** codes are not included.

| Code of <err> | Meaning |
|---------------|-----------------------------|
| 0 | phone failure |
| 1 | no connection to phone |
| 2 | phone-adaptor link reserved |
| 3 | operation not allowed |
| 4 | operation not supported |
| 5 | PH-SIM PIN required |
| 6 | PH-FSIM PIN required |
| 7 | PH-FSIM PUK required |
| 10 | SIM not inserted |
| 11 | SIM PIN required |
| 12 | SIM PUK required |
| 13 | SIM failure |
| 14 | SIM busy |
| 15 | SIM wrong |
| 16 | incorrect password |
| 17 | SIM PIN2 required |
| 18 | SIM PUK2 required |
| 20 | memory full |
| 21 | invalid index |

| | |
|------|---|
| 22 | not found |
| 23 | memory failure |
| 24 | text string too long |
| 25 | invalid characters in text string |
| 26 | dial string too long |
| 27 | invalid characters in dial string |
| 30 | no network service |
| 31 | network timeout |
| 32 | network not allowed - emergency calls only |
| 40 | network personalization PIN required |
| 41 | network personalization PUK required |
| 42 | network subset personalization PIN required |
| 43 | network subset personalization PUK required |
| 44 | service provider personalization PIN required |
| 45 | service provider personalization PUK required |
| 46 | corporate personalization PIN required |
| 47 | corporate personalization PUK required |
| 103 | illegal MS |
| 106 | illegal ME |
| 107 | GPRS services not allowed |
| 111 | PLMN not allowed |
| 112 | location area not allowed |
| 113 | roaming not allowed in this location area |
| 132 | service option not supported |
| 133 | requested service option not subscribed |
| 134 | service option temporarily out of order |
| 148 | unspecified GPRS error |
| 149 | PDP authentication failure |
| 150 | invalid mobile class |
| 302 | Command conflict |
| 1000 | PAD mode TCP idle timeout |
| 1001 | PAD mode failed to establish TCP connection |

| | |
|------|---------------------------------------|
| 1002 | PAD mode failed to get IP address |
| 1003 | PAD mode peer disconnect the session |
| 3513 | Unread records on SIM |
| 3516 | Couldn't read SMS parameters from SIM |
| 3517 | SM not ready |
| 3518 | Invalid parameter |
| 3738 | CSCS mode not found |
| 3742 | CPOL operation format wrong |
| 3765 | Invalid input value |
| 3769 | Unable to get control |
| 3771 | Call setup in progress |
| 3772 | SIM powered down |

CMS ERROR Codes

Final result code **+CMS ERROR: <err>** indicates an error related to mobile equipment or network. The operation is similar to ERROR result code. None of the following commands in the same command line is executed. Neither **ERROR** nor **OK** result code shall be returned.

<err> values are mostly used by common messaging commands:

| Code of <err> | Meaning |
|---------------|-------------------------|
| 300 | ME failure |
| 301 | SMS ME reserved |
| 302 | operation not allowed |
| 303 | operation not supported |
| 304 | invalid PDU mode |
| 305 | invalid text mode |
| 310 | SIM not inserted |
| 311 | SIM pin necessary |
| 312 | PH SIM pin necessary |
| 313 | SIM failure |
| 314 | SIM busy |

| | |
|-----|----------------------------------|
| 315 | SIM wrong |
| 316 | SIM PUK required |
| 317 | SIM PIN2 required |
| 318 | SIM PUK2 required |
| 320 | memory failure |
| 321 | invalid memory index |
| 322 | memory full |
| 330 | SMSC address unknown |
| 331 | no network |
| 332 | network timeout |
| 500 | unknown |
| 512 | SIM not ready |
| 513 | message length exceeds |
| 514 | invalid request parameters |
| 515 | ME storage failure |
| 517 | Invalid service mode |
| 528 | more message to send state error |
| 529 | MO SMS is not allow |
| 530 | GPRS is suspended |
| 531 | ME storage full |
| 532 | doing SIM refresh |

Summary of URC

| Index | URC display | Meaning | Condition |
|-------|--|--|-------------|
| 1 | +CMTI:<mem>,<index> | New message received, and saved to memory | AT+CNMI=2,1 |
| 2 | +CMT:[<alpha>],<length><CR><LF>< pdu> | New short message is received and output directly to TE(PDU mode) | AT+CNMI=2,2 |
| 3 | +CMT:<oa>,<alpha>,<scts>,<tooa>,<foo>,<pid>,<dcs>,<sca | New short message is received and output directly to TE(Text mode) | AT+CNMI=2,2 |

| | | | |
|----|---|--|---------------------|
| | >,<tosca>,<length>] <CR><LF><data> | | |
| 4 | +CBM:<length><CR> > | New CBM is received and output directly(PDU mode) | AT+CNMI=2,2 |
| 5 | +CBM:<sn>,<mid>,< dcs>,<page>,<pages>,<CR>,<LF><data> | New CBM is received and output directly to TE(Text mode) | AT+CNMI=2,2 |
| 6 | +CDS:<length><CR> <LF><pdu> | New CDS is received and output directly(PDU mode) | AT+CNMI=2,2 |
| 7 | +CDS:<fo>,<mr>,[<ra>],[<tora>],<scts>,<dt>,<st> | New CDS is received and output directly to TE(Text mode) | AT+CNMI=2,2 |
| 8 | +CGEV:NW DETACH | GPRS network detach | AT+CGEREP=1 |
| 9 | +CGEV:ME DETACH | GPRS ME detach | AT+CGEREP=1 |
| 10 | +CGREG:1 | Network registered | AT+CGREG=1 |
| 11 | +CGREG:0 | Network unregistered | AT+CGREG=1 |
| 12 | +CGREG:1,<lac><ci> | Network registered, with location code | AT+CGREG=2 |
| 13 | +CGREG:0,<lac><ci> | Network unregistered, with location code | AT+CGREG=2 |
| 14 | +QCGTIND | A CS voice call, CS data, fax call or GPRS session termination indicator | AT+QCGTIND=1 |
| 15 | +CSQN:<rsqi>,<ber> | Signal quality change | AT+QEXTUNSOL="SQ",1 |
| 16 | +CMWT:<store>,<index>,<voice>,<fax>,<email>,<other> | Message waiting | AT+QEXTUNSOL="MW",1 |
| 17 | +QGURC:<event> | Unsolicited result code follow particular call state transition | AT+QEXTUNSOL="UR",1 |
| 18 | +CBCN<bcs>,<bcl> | Display battery connection status and battery charge level | AT+QEXTUNSOL="BC",1 |
| 19 | +QBAND:<band> | Band mode display | AT+QEXTUNSOL="BM",1 |

| | | | |
|----|--|--|---------------------|
| 20 | +TSMSINFO:<CMS error info> | Additional SMS information | AT+QEXTUNSOL="SM",1 |
| 21 | +CCINFO:<Call is Disconnected>,<remain calls> | Displays the disconnected call ID and the remain call numbers after one of the call disconnected | AT+QEXTUNSOL="CC",1 |
| 22 | RING | Indicates incoming call | n/a |
| 23 | Call Ready | Device ready to make/receive calls | n/a |
| 24 | +QTEMP:-1 | Low temperature warning | AT+QTEMP=1 |
| 25 | +QTEMP:1 | High temperature warning | AT+QTEMP=1 |
| 26 | +QTEMP:-2 | Low temperature shutdown indicator | AT+QTEMP=1 |
| 27 | +QTEMP:2 | High temperature shutdown indicator | AT+QTEMP=1 |
| 28 | UNDER_VOLTAGE POWER DOWN | Under voltage shutdown indication | n/a |
| 29 | UNDER_VOLTAGE WARNING | Under voltage warning | n/a |
| 30 | OVER_VOLTAGE POWER DOWN | Over voltage shutdown indication | n/a |
| 31 | OVER_VOLTAGE WARNING | Over voltage warning | n/a |
| 32 | UNDER_VOLTAGE POWER DOWN | Normal power down | n/a |
| 33 | +COLP:<number>,<type>[,<subaddr>,<statype>[CLI validity]], | The presentation of the COL(connected line) at the TE for a mobile originated call | AT+COLP=1 |
| 34 | +CLIP:<number>,<type>"" ,,<alphaID>,<CLI validity> | Mobile terminating call indication | AT+CLIP=1 |
| 35 | +CRING:<type> | An incoming call is indicated to the TE with unsolicited result code instead of the normal RING | AT+CRC=1 |
| 36 | +CREG:<stat> | Indicate registration status of the ME | AT+CREG=1 |

| | | | |
|----|---|--|--------------|
| 37 | +CREG:<stat>[,<lac>]] | After cell neighborhood changing shows whether the network has currently indicated the registration of the ME, with location area code | AT+CREG=2 |
| 38 | CCWV | Call meter warning,5 seconds left before ACM | AT+CCWV=1 |
| 39 | +CCWA:<number>,<type>,<class>[,<alpha>] | Call waiting indication | AT+CCWA=1,1 |
| 40 | RDY | ME initialization successful | n/a |
| 41 | +CFUN:1 | All function of the ME is available | n/a |
| 42 | +CPIN:<state> | SIM card pin state | n/a |
| 43 | MO RING | MO call ringing | AT+QMOSTAT=1 |
| 44 | MO CONNECTED | MO call connected | AT+QMOSTAT=1 |
| 45 | ALARM RING | Alarm event triggered | |

Index

| | |
|-------------------------------|--------------------------------|
| AT\$AUXUARTCFG, 49 | AT\$TTFNT, 40, 41, 75, 84, 144 |
| AT\$CGPCO, 32 | AT\$TTFNTDEL, 41, 77, 84 |
| AT\$RESET, 91 | AT\$TTFNTDELA, 77 |
| AT\$TTACCAM, 116 | AT\$TTFNTDELR, 78 |
| AT\$TTACKCFG, 46 | AT\$TTFNTEXE, 94 |
| AT\$TTARG, 26, 27, 28, 69 | AT\$TTFNTQRY, 82 |
| AT\$TTATPW, 29 | AT\$TTFTPCD, 58 |
| AT\$TTBUZCFG, 118 | AT\$TTFTPCL, 59 |
| AT\$TTBUZCTR, 95, 96, 97, 119 | AT\$TTFTPCN, 57 |
| AT\$TTCAIDLST, 74 | AT\$TTFTPDEF, 57 |
| AT\$TTCNT, 80 | AT\$TTFTPDIR, 59 |
| AT\$TTCNTQRY, 81 | AT\$TTFTPOP, 58, 61 |
| AT\$TTDEVID, 31 | AT\$TTFTPRST, 63 |
| AT\$TTDEVSTAT, 88 | AT\$TTFTPRT, 60 |
| AT\$TTDEVTYP, 88 | AT\$TTFTPSTR, 62 |
| AT\$TTDI, 120 | AT\$TTGEOFNC, 105, 136 |
| AT\$TTDL, 121 | AT\$TTGFDB, 107 |
| AT\$TTECHO, 20 | AT\$TTGFDEL, 108 |
| AT\$TTEVGQRY, 79 | AT\$TTGFIND, 106 |
| AT\$TTFFA, 54 | AT\$TTGPSHEAD, 114 |

TT8750+ AT Command Set

AT\$TTGPSOSI, 152, 158, 160, 164

AT\$TTGPSPORT, 99

AT\$TTGPSQRY, 100

AT\$TTGPSRI, 101

AT\$TTGPSSTT, 98

AT\$TTGSMJDC, 72

AT\$TTHB, 121

AT\$TTHC, 122

AT\$TTICCID, 85

AT\$TTIGNDB, 24

AT\$TTIOCO, 24

AT\$TTIOCO(x), 24

AT\$TTIODB, 23

AT\$TTIPHB, 123

AT\$TTIW, 123

AT\$TTLOGCL, 34

AT\$TTLOGDMP, 35

AT\$TTLOGRD, 36

AT\$TTMAPLINK, 93

AT\$TTMGR, 18

AT\$TTMSGMASK, 33

AT\$TTNETIP, 28, 89

AT\$TTNETWD, 71

AT\$TTNR, 124

TT8750+

AT\$TTNRCFG, 21

AT\$TTODOENA, 104

AT\$TTODOM, 102

AT\$TTOVSDEF, 104

AT\$TTPADCMF, 52

AT\$TTPADDST, 38

AT\$TTPASP, 26

AT\$TTPASPACT, 27

AT\$TTPASPCFG, 50

AT\$TTPKG, 89

AT\$TTPLYDEL, 111

AT\$TTPLYFN, 109

AT\$TTPLYGFIND, 110

AT\$TTPLYGFN, 109

AT\$TTPWSRV, 92

AT\$TTRBI, 129

AT\$TTRI, 124

AT\$TTRN, 125

AT\$TTRROUTE, 112

AT\$TTRROUTE, 112

AT\$TTRROUTE, 112

AT\$TTRROUTEDEL, 114

AT\$TTRROUTEIND, 113

AT\$TTRR, 126

TT8750+ AT Command Set

| | |
|---------------------------------|----------------------------|
| AT\$TTRTCAL, 65 | AT\$TTUSRVAL, 85 |
| AT\$TTRTCTI, 66, 68 | AT\$TTWAKEUP, 40 |
| AT\$TTSDT, 19 | AT%BYCNT, 41 |
| AT\$TTSFOTA, 53 | AT%CGAATT, 69 |
| AT\$TTSI, 126 | AT%CGREG, 71, 89, 136, 147 |
| AT\$TTSMSDST, 38, 39 | AT%DM, 86 |
| AT\$TTSNDMG, 37, 38 | AT&F, 35, 37, 103 |
| AT\$TTSP, 120 | AT&W, 76, 80 |
| AT\$TTSR, 127 | AT+CACM, 165 |
| AT\$TTSRN, 91 | AT+CCWA, 165 |
| AT\$TTSRVDST, 30, 38, 41 | AT+CEER, 90 |
| AT\$TTSTOCMD, 38, 83 | AT+CGACT, 89 |
| AT\$TTSVL, 19 | AT+CGDCONT, 69 |
| AT\$TTTA, 128 | AT+CGREG, 166 |
| AT\$TTTCPAPI, 43, 46 | AT+CLAE, 165 |
| AT\$TTTCPCFG, 44 | AT+CLIP, 165 |
| AT\$TTTCPCNT, 45 | AT+CMEE, 165 |
| AT\$TTTCPRST, 45, 46 | AT+CMGS, 16 |
| AT\$TTTE, 128 | AT+CNMI, 166 |
| AT\$TTTODOM, 151, 157, 160, 163 | AT+COLP, 165 |
| AT\$TTTOLST, 70 | AT+COPS, 89 |
| AT\$TTTRGEV, 141 | AT+CPIN, 69, 89 |
| AT\$TTTRGEV, 79 | AT+CR, 165 |
| AT\$TTTRODOM, 103 | AT+CRC, 16, 165 |

TT8750+ AT Command Set

AT+CREG, 89, 136, 147, 165

AT+CSMP, 166

AT+CSQ, 89

AT+CSSN, 165

AT+CUSD, 166

ATD, 26, 27

ATI, 16

ATO, 27