

Terminus Tracker Products User Manual



JANUS REMOTE
C ● M M U N I C A T I O N S ●

Bulletin	JA14-UM
Revision	A00
Date	30 Oct 2012

TABLE OF CONTENTS

TABLE OF CONTENTS and DISCLAIMER.....	2
1 APPLICABILITY TABLE.....	3
2 REFERENCES	3
2.1 Telit Document List	
2.2 Janus Document List	
3 OVERVIEW.....	3
3.1 Introduction.....	3
3.2 Mechanical Diagram.....	3
4 INTERFACES	4-6
4.1 I/O Connector	4
4.2 Pin-Out	4
4.3 Signal Detail	5-6
5 TECHNICAL SPECIFICATIONS.....	7
5.1 Mechanical Specifications	7
5.2 Electrical Specifications.....	7
6 AT Command Guide.....	8
6.1 AT Command Format	8
6.2 AT Command Example.....	8
7 MODEM OPERATION.....	9-11
7.1 Setting up a terminal emulator for use with the Terminus Tracker	9
7.2 Powering ON/OFF	9
7.3 Setting up Service - Network Settings.....	9
7.4 Sending an SMS.....	10
7.5 Making a GPRS Data Call	10
7.6 Acquiring GPS Data	11
8 PYTHON	12-14
8.1 What is Python	12
8.2 Telit's Python Implementation	12
8.3 Python Development Tools	12
8.4 Installing the IDE	12
8.5 Patching the Install	12
8.6 Setting up the IDE	13
8.7 Python Interpreter.....	14
9 TRACKER DEMO	15-22
9.1 Overview.....	15
9.2 Demo Package Contents.....	15
9.3 Tracker Demo Checklist.....	15
9.4 Installing the Tracker Demo	15
9.5 Configuring the Demo.....	15-18
9.6 Uploading the Demo.....	18-20
9.7 Setting the Script to Auto Start	20
9.8 Opening the Demo GUI	20
9.9 Running the Demo.....	21
9.10 SMS on the Fly Configurations.....	22
ORDER INFORMATION and REVISION HISTORY	23

DISCLAIMER

The information contained in this document is the proprietary information of Connor-Winfield Corporation and its affiliates (Janus Remote Communication). The contents are confidential and any disclosure to persons other than the officers, employees, agents or subcontractors of the owner or licensee of this document, without the prior written consent of Connor-Winfield, is strictly prohibited. Connor-Winfield makes every effort to ensure the quality of the information it makes available. Notwithstanding the foregoing, Connor-Winfield does not make any warranty as to the information contained herein, and does not accept any liability for any injury, loss or damage of any kind incurred by use of or reliance upon the information. Connor-Winfield disclaims any and all responsibility for the application of the devices characterized in this document, and notes that the application of the device must comply with the safety standards of the applicable country, and where applicable, with the relevant wiring rules. Connor-Winfield reserves the right to make modifications, additions and deletions to this document due to typographical errors, inaccurate information, or improvements to programs and/or equipment at any time and without notice. Such changes will, nevertheless be incorporated into new editions of this application note.

All rights reserved 2012 Connor-Winfield Corporation

1 APPLICABILITY TABLE

Product	Part Number
Terminus Tracker	v1.0

2 REFERENCES

2.1 Telit Document List

GE864-GPS Hardware User Guide

GE864-GPS Software User Guide

AT Command Reference

Easy GPRS User Guide

Easy Script Python

2.2 Janus Document List

Please refer to the NavSync's website, www.navsync.com, for the latest information on the MS20.

MS20 Documentation

3 OVERVIEW

3.1 Introduction

The Terminus Tracker is a self contained, quad-band GSM/GPRS device designed to provide an inexpensive and comprehensive solution to asset tracking. It utilizes the proven technology of the Telit GE864-GPS module for its core communications engine. The embedded Python allows for ultimate flexibility in autonomous functionality for the user's specific application.

3.2 Mechanical Diagram

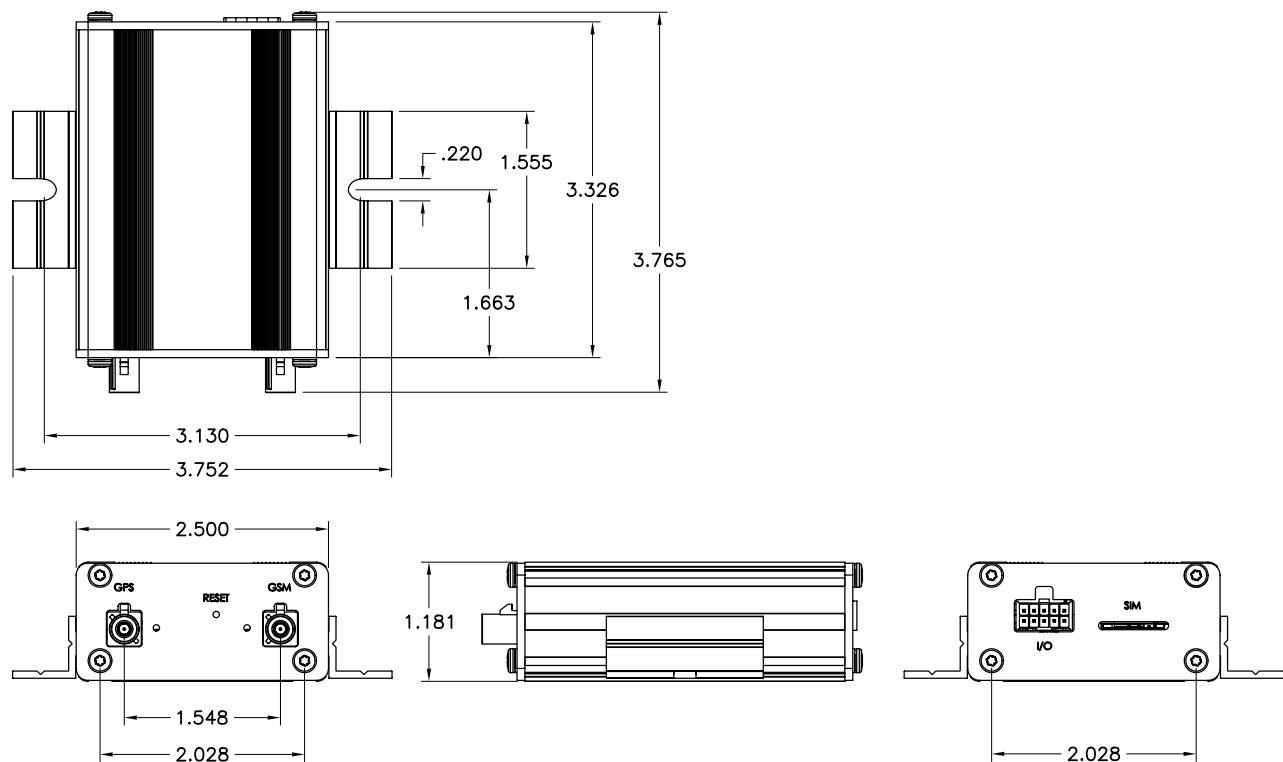


Figure 1 Terminus Tracker Mechanical Diagram

4 Interfaces

4.1 I/O Connector

The interface of the Terminus Tracker is a 2x5 locking header for rugged operation as well as allowing custom harnesses to be made.

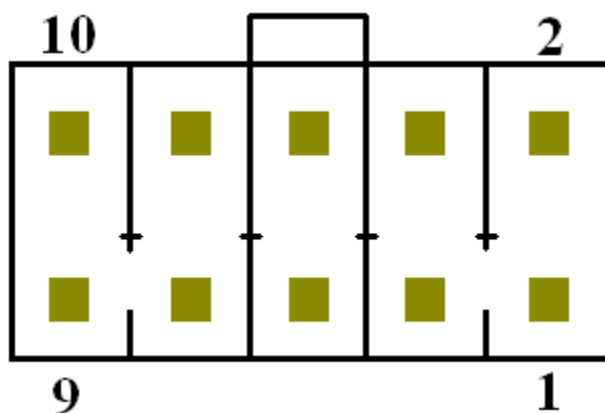


Figure 2 Depiction of pin numbers and orientation on the tracker

4.2 Pin Out

Pin	Description	Direction	Level	Note
1	N/O Sensor	Input	Supply	
2	Supply Enable	Input	Supply	
3	Modem ON/OFF	Input	2.8v CMOS	
4	Active High Sensor	Input	Supply	
5	Received Data (RX)	Output	RS-232	
6	Transmitted Data (TX)	Input	RS-232	
7	Request to Send (RTS)	Input	RS-232	
8	Clear to Send (CTS)	Output	RS-232	
9	Supply – (GND)	N/A	Ground	
10	Supply + (VCC)	N/A	Supply	

Figure 3

Samtec Part Number:
Housing: IPD1-5-D-K

Janus Store Part Number:
Housing: XT-507-G

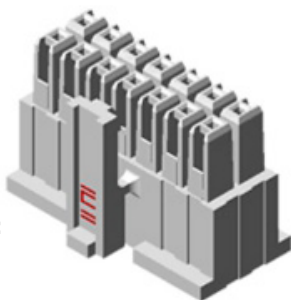


Figure 4

Samtec Part Number:
Contacts: IPD1-5-D-K

Janus Store Part Number:
Contacts: XT-507-G



4 Interfaces continued

4.3 Signal Detail

4.3.1 Power

The Terminus Tracker accepts input voltages from 6 to 17VDC during normal operation. It contains a switching regulator with >80% efficiency, making the power draw very manageable for automotive battery connections. The supply does not have reverse voltage protection.

4.3.2 Serial Communications

The serial interface is an RS-232 port that supports both EIA/TIA-232 and V.28/V.24 communications protocols. The interface supports RTS/CTS handshaking, and the following are default settings:

- Baud Rate: 115200bps
- Bits: 8
- Stop Bits: 1
- Parity: None
- Hardware Handshaking: Yes

Although 115200bps is the default, the GE864-GPS modem does support auto-baud when communicating to the DTE for AT commands.

4.3.3 Isolated Inputs

The Terminus Tracker has 2 basic sensor inputs it can monitor and report on:

- The N/O Sensor is a normally open input with a 15k ohm pull up to Vin. This input is designed to be open collector, or switch controlled (i.e. a closing door).
- The Active High Sensor is configured with a 15k ohm inline resistor. This input is designed to be actively driven between 6-17VDC (i.e. an ignition signal).

These inputs can be left floating if unused.

4.3.4 Modem ON/OFF

This signal can be used to turn the cellular modem ON and OFF. This signal must be controlled via open collector or switch, do not use a pull up.

Pull LOW for ~3 seconds then release to turn the unit ON or OFF.

4.3.5 Supply Enable

This is an enable signal for the on board regulator, pull to ground to disable the regulator. Control via open collector or switch, do not use a pull up, and leave floating if unused. This signal is not intended to be used as a means of turning off the Terminus Tracker, use the ON/OFF signal otherwise.

4.3.6 SIM

The SIM Card interface is a push push recessed position that will accept the subscriber card provided by the provider. It can accommodate a 1.8v or 3.0v SIM card.

4.3.7 Reset

The Reset button is connected to the reset signal of the GE864-GPS. This signal is intended only as a method to reset the unit if it becomes unresponsive. It is not intended to be used as a means of turning the Terminus Tracker OFF, use the ON/OFF signal otherwise.

4 Interfaces continued

4.3 Signal Detail continued

4.3.8 LEDs

The Terminus Tracker has two LEDs available, a cellular status LED and a user LED. The cellular LED tracks the status of the modem, whereas the user LED is controlled by adjusting GPIO #5.

Cellular Status LED (Amber)

LED Status	Cellular Status
Permanently OFF	Cellular Modem OFF
Fast Blinking (0.5s ON/OFF)	Searching/Registering
Slow Blinking (0.3s ON/2.7s OFF)	Registered
Permanently ON	A call is active

User LED (Green)

LED Status	Status
Permanently OFF	User Controlled
Permanently ON	User Controlled

The Cellular Status LED is controlled via AT#SLED command, the User LED is controlled via AT#GPIO=5 command.

4.3.9 Cellular/GPS Antenna Connections

The RF interfaces on the Terminus Tracker are geared with the automotive industry in mind, and are available via FAKRA plugs. The GPS interface can supply a bias voltage of 2.8VDC to an active antenna, or be turned off if the antenna cable is short and the internal LNA is sufficient.

GSM Antenna

Interface Type : FAKRA D (Bordeaux Violet)

GSM Antenna Specifications

Frequency Range	Depending by frequency band(s) provided by the network operator, the customer shall use the most suitable antenna for that/those band(s)
Bandwidth	70Mhz (GSM850) 80Mhz (GSM900) 170Mhz (DCS) 140Mhz (PCS)
Gain	<1.4dBi (GSM850 & GSM900)
Impedance	50 Ohm
Input Power	> 2W peak
VSWR (Max)	<= 10:1 (limit to avoid permanent damage)
VSWR (Recommended)	<= 2:1 (limit to fulfill all regulatory requirements)

GPS Antenna

Interface Type: FAKRA C (Blue)

GPS Antenna Specifications

Input Voltage	2.8Vdc
Frequency Range	1578.42 +/- 3Mhz
Gain	<20dB total at the connector
Impedance	50 Ohms

If you are not using an active antenna and can take advantage of the internal LNA, AT\$GPSAT can be used to disable the bias voltage and turn the LNA on. The internal LNA provides a gain of 18dB when active. The bias voltage is only active when the internal LNA is disabled to avoid over saturation.

5 Technical Specifications

5.1 Mechanical Specifications

Currently N/A

5.2 Electrical Specifications

5.2.1 Absolute Maximum Ratings

Parameter	Min	Typ	Max	Unit	Note
Vin (Supply)	0	-	40	Vdc	
Vin (Isolated Inputs)	0	-	100	Vdc	
Storage Temperature	-40	-	TBD	°C	

Operation of the device at these or any other conditions beyond those listed under the Recommended Operating Conditions is not implied. Exposure to Absolutely Maximum Rating conditions for extended periods of time may affect device reliability.

5.2.2 Recommended Operating Conditions

Parameter	Min	Typ	Max	Unit	Note
Vin (Supply)	6	12	17	Vdc	
Vin (Isolated Inputs)	0	12	17	Vdc	
Temperature	-30	-	80	°C	

5.2.3 Recommended Operating Conditions – Supply Enable

Parameter	Min	Typ	Max	Unit	Note
Input Voltage High - VIH	1.26	-	-	Vdc	
Input Voltage LOW - VIL	-	-	1.13	Vdc	

5.2.4 Electrical Characteristics

Parameter	Min	Typ	Max	Unit	Note
Input Voltage	6	12	17	Vdc	

Average Current – GPS OFF

Parameter	Min	Typ	Max	Unit	Note
Modem OFF	6	4	3.5	mA	
Registered, Idle	20	11	9	mA	
Power Save (CFUN=0)	10	7	5	mA	
GPRS Data Call	60	70	130	mA	

Average Current – GPS ON

Parameter	Min	Typ	Max	Unit	Note
Modem OFF	6	4	3.5	mA	
Registered, Idle	60	34	24	mA	
Power Save (CFUN=0)	53	30	20	mA	3
GPRS Data Call	75	120	240	mA	

1. All numbers are average, taken at 25°C.
2. Numbers taken with RS232 active, adding ~3mA current draw. Remove serial connection externally to disable RS232.
3. Although noted, when using CFUN=0 mode, it's recommended to turn the GPS OFF to maximize draw reduction.

6 AT Command Guide

The basic AT commands used to control the modem's operation are defined in this section. Please refer to Telit AT Command Guide for detailed command definition.

6.1 AT Command Format

A command line is a string of characters sent from a DTE or emulation software to the Terminus while the modem is in a command state. A command line has a prefix, a command, and a terminator. Each command line must begin with the character sequence AT and must be terminated by a carriage return and line feed (ie ASCII Carriage Return <13> and ASCII LineFeed <10>).

Commands entered in upper case or lower case are accepted, but both the A and T must be of the same case, i.e., "AT" = ASCII 065, 084 or "at" = ASCII 097, 116. The command is a string of commands restricted to printable ASCII characters (032 - 126). Standardized GSM commands start with a "+" character and proprietary commands can start with one of the following characters, "@", "#", "\$" or "*". Space characters (ASCII 032) in a command are ignored and control characters other than defined terminator characters CR (ASCII 013) and LF (ASCII 010) in the command string will cause an ERROR result code.

The default terminator is the ASCII <CR> character followed by the ASCII <LF> character. The command line interpretation begins upon receipt of the carriage return and line feed characters. If a syntax error is found anywhere in a command line command, the remainder of the line will be ignored and the ERROR result code will be returned. Most commands entered with parameters out of range will not be accepted and the ERROR response will be returned to the DTE or emulation software.

Commands will only be accepted by the modem once the previous command has been fully executed, which is normally indicated by the return of an appropriate result code.

6.2 AT Command Example

Example AT Command

AT#BND=3<cr><lf>	-Select default band
AT&P0<cr><lf>	-Select Profile 0
AT&W0<cr><lf>	-Save Profile 0

7 Modem Operation

7.1 Setting up a terminal emulator for use with the Terminus Tracker

To interface with the tracker, connect the RS-232 serial interface to a PC and use a terminal emulation program such as Microsoft Hyperterminal. Set the following parameters:

- Baud Rate: 115200bps
- Bits: 8
- Stop Bits: 1
- Parity: None
- Hardware Handshaking: Yes

7.2 Powering ON/OFF

The Terminus Tracker has a couple of ways to be turned ON. The default configuration is automatic turn on that is controlled by an on-board microcontroller. This can be changed by running the python script JANUS_IO.AutoONControl. See the script for details.

- Auto-on: The unit will automatically turn ON when power is applied, regardless of the monitored inputs or the ON/OFF signal.
- Event Based: The unit will remain OFF and in standby until the monitored inputs are triggered or the ON/OFF signal is pulled LOW for ~3 seconds then released.

There are two ways to switch OFF the tracker. These can only be successfully used when the unit is in Event Based mode, otherwise it will simply turn back on.

- Use the AT command AT#SHDN
- Use the ON/OFF signal and pull it LOW for ~3s then release.

7.3 Setting up Service – Network Settings

The network settings for the Terminus Tracker will vary depending on the carrier being used. Below are two of the North American cases for these settings.

For AT&T & AT&T MVNO (Jasper, Kore, Aeris, Qualcom-nPhase, CrossBridge)

```
AT#SELINT=2 //use of most recent AT command set
AT#ENS=1 //AT&T configuration
AT#AUTOBND=2 //enable Quad band system selection (optional)
AT&P0 //select profile
AT&W0 //save setting
```

For T-Mobile, ROGERS, and other non-AT&T MVNOs (Jasper, Wyleless, Raco, Sensor Logic)

```
AT#SELINT=2 //use of most recent AT command set
AT#STIA=2, 10 //enable SAT-SIM Application Tool-Kit
AT# BND=3 //default bands to 850/1900
AT#AUTOBND=2 //enable quad band system selection (optional)
AT#PLMNMODE=1 //enable EONS (enhanced operator naming scheme)
AT&P0 //save profile
AT&W0 //save setting
```

If the Terminus Tracker is being used in a different country or with a different carrier please refer to the Telit AT Command reference document regarding the use of the AT#BND command to set the proper frequency band.

Important: After entering either one of the sets of settings above, power the unit OFF then ON. It is now ready for usage.

7 Modem Operation continued

7.3 Setting up Service – Network Settings continued

Check the network status

- Ensure a valid SIM card is installed
- *Enter AT+CREG?<cr> and wait for response
 - Response should be +CREG: 0,1 meaning the device is registered on the home network, or +CREG: 0,5 meaning the device is registered but roaming.
- *Enter AT+CGREG?<cr> and wait for response
 - Response should be +CGREG: 0,1 or +CGREG: 0,5 meaning a data connection is available.
- Enter AT+CSQ<cr> and wait for response.
 - Response should be +CSQ:<rssi>,<ber> which denotes the signal quality.

rssi	Signal Strength
99	Not known or not detectable
0-31	$\text{dBm} = (\text{rssi} * 2) - 113$

Note: A result of 31 indicates -51dBm or greater.

7.4 Sending an SMS

You can send and receive SMS messages through the following methods.

7.4.1 Configuration:

- Enter AT+CMGF=1<cr> to set text mode.

7.4.2 Sending:

- Enter AT+CMGS=12345678910
- Wait for prompt ">" then enter the message to send
- Enter "CTRL+Z"<cr> to send the message

7.4.3 Receiving:

- Enter AT+CNMI=2,1 to set the message indication
- Wait for indication of +CMTI: <memr>, <index>
- Enter AT+CMGL=ALL to view all received messages
- Enter AT+CMGD=1,4 to delete all stored messages

7.5 Making a GPRS Data call

GPRS is a data service that uses Packet Data Protocol (PDP).

- Set up the PDP context parameters
 - Enter AT+CGDCONT=1, "IP", "APN", "0.0.0.0",0,0<cr>
 - Where APN is specific to the service provider being used.
- Activate the PDP context
 - Enter AT#SGACT=1,1,"v", "p"
 - Where v is your user ID and p is your password.
 - If these are not set replace with "", ""
 - Enter AT#SD=1,0,IPP,IPA,0,0,0
 - Look for response "CONNECT". This opens a remote connection via socket
 - IPP = the remote host port of the server you are trying to connect to. (0 to 65535)
 - IPA = the IP address of the server you are trying to connect to in the format: "xxx.xxx.xxx.xxx"
 - 'Port' = the remote host port to contact provided by carrier (0 to 65535)
- At this point a data session is active and data can be sent from the Terminus to the remote device and visa versa.
- To exit the data session and return to command mode, send the characters "+++" and wait for the OK response
- Enter AT#SH=1 to close the socket

7 Modem Operation continued

7.6 Acquiring GPS Data

The GPS data can be retrieved by the following methods.

7.6.1 Configuration:

- Enter AT\$GPSD=2<cr> to put the unit in controlled mode, this is required for the Terminus Tracker
- Configure the antenna gain:
 - If you have an active antenna: Enter AT\$GPSAT=1 to activate the voltage bias and turn OFF the internal LNA
 - If you have a short cable passive antenna: Enter AT\$GPSAT=0 to turn off the voltage bias and turn ON the internal LNA.
- Enter AT\$GPSP=1 to turn the GPS ON.

7.6.2 Retrieving the Data:

- Enter AT\$GPSACP to retrieve the currently acquired position
- Enter AT\$GPSNMUN=2,1,1,1,1,1,1 to begin a stream of NMEA data with all sentences active. The particular sentences to be reported can be turned on and off depending on what is set on this command.

See the AT Command Guide for further details on these commands.

8 Python

8.1 What is Python?

First released by Guido van Rossum in 1991, Python is a clear and powerful dynamic object-oriented programming language used in a wide variety of application domains. It offers strong support for integration with other languages and tools, comes with extensive standard libraries, and can be learned in a few days. Python versions are available for Windows™, Linux/Unix™, Mac™ OS X, OS/2, Amiga™, Palm Handhelds, and other platforms. It has an open source license that makes it free to use and distribute, even for commercial applications.

8.2 Telit's Python Implementation

The ability to create and execute embedded applications is the heart of the Terminus Tracker, which gives it the ability to be autonomous and flexible to the user's application. Telit chose the Python scripting language to implement this functionality and produced a complete software platform called Telit Easy Script to enable the development of M2M applications. Using Telit Easy Script allows the customer to easily create, install and execute programs on their Telit modules. The Telit Easy Script implementation offers:

- Python script interpreter engine
- 2MB of non-volatile memory for user scripts and data
- 1.2MB of RAM reserved for Python engine usage
- 16KB of memory for each variable

Telit developed the modules with built-in interfaces, which provide communication between the Python script and the Telit module. The interfaces are dedicated to M2M applications. Refer to the Telit Easy Script in Python document for a detailed discussion of these modules.

The Telit Python implementation uses version 1.5.2+ which includes use of all Python statements and almost all built-in types and functions (see Telit document for details). The plus on the version number indicates the addition of version 2.0 string handling methods. Please refer to Telit Document: Easy Script in Python.

8.3 Python Development Tools

The Telit Python package (available at www.janus-rc.com) includes an integrated development environment (IDE) to make developing scripts more straightforward. This IDE includes a full editor, an interactive debugger and a compiler.

8.4 Installing the IDE

Execute the file TelitPy1.5.2+_V4.1.exe to install Telit Python package and let the installer use the default settings. The Python compiler package is contained in the installation. The Telit Python package is placed in the folder C:\Program Files\Python\. The correct path in the Windows Environmental variables will be set up automatically.

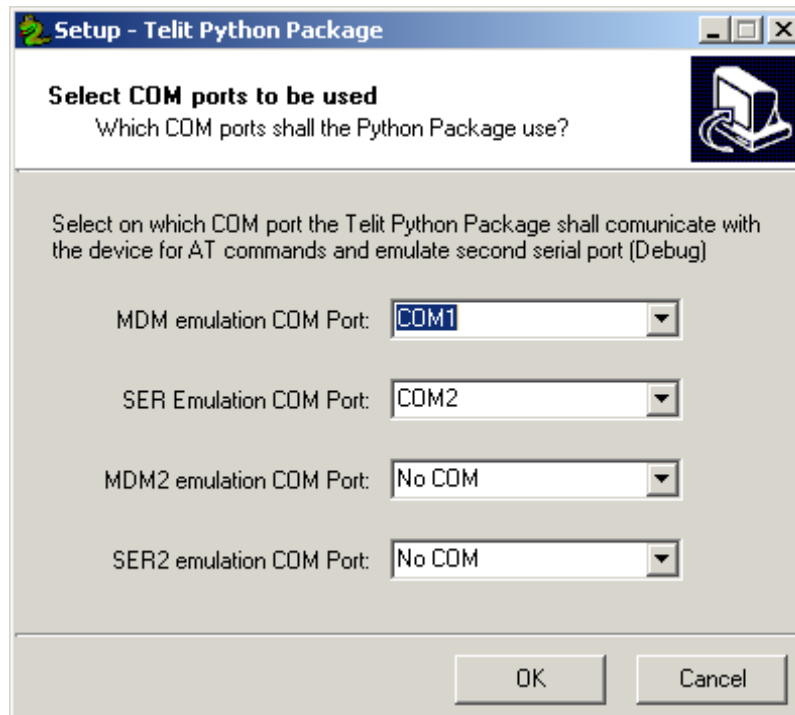
8.5 Patching the install

In order to take advantage of the Right Click -> Download script ability of the IDE, it must be patched with the included adjusted file "serialwin32.py". Simply copy the included .py and .pyo files and paste them into the C:\Program Files\Python\Lib folder, overwriting the old ones.

8 Python continued

8.6 Setting up the IDE

The only required setup is to configure the COM ports with the COM Port Selection Tool as shown.



MDM emulation COM Port: This is the only port that must be assigned. The MDM assignment is used for the following tasks:

1. When executing script from IDE the IDE sends and receives AT commands sent via the MDM module. Example: `res = MDM.send('AT+CREG?',0)` The command `AT+CREG?` Would be sent to the Terminus via COM1. This means that the Terminus must be connected to COM1.
2. Script files are downloaded to module from the IDE or Windows File Explorer via COM1.

All other Modules SER, MDM2 and SER2 are the COM ports that will need to be assigned if you use the IDE to run your script and you have used these modules in your code.

8 Python continued

8.7 Python Interpreter

Python scripts are text files stored in NVM inside the Telit module. The file system inside the module can write and read files. The Python script is executed at the lowest priority to make sure this does not interfere with GSM/GPRS normal operations. Four steps are required to create a script for controlling the Terminus via the Python engine. These are: Write the Python script; Compile and download the Python script into the module's NVM; Enable the Python script; and Execute the Python script.

- Write Python script
 - You can write and test the Python script directly from the IDE. Refer to section 3.1 of the Telit Easy Script in Python document for a short example.
- Compile and Download Python script
 - When the Pythonwin IDE is installed on the PC it adds the file types .py and .pyo to the Windows file systems and associates them with the Pythonwin tools. It also installs two programs, a compiler and a file loader, that are then accessible from the Windows file menu. This makes working with Python script files fairly easy.
 - To compile a script just right-click on the .py filename from Windows Explorer and select compile from the pop-up menu. This then creates the .pyo file. Multiple files can be compiled at the same time by selecting them all, then selecting compile.
 - To download the compiled files from the computer to the Terminus, just select the .pyo file(s) in Windows Explorer and right-click. Select download from the pop-up menu. The compiled Python script(s) will be moved into the Terminus' NVM.
- Enable Python script
 - To enable the Python script the Terminus script mode must first be set by sending the AT#STARTMODESCR=1,10 command. This tells the Terminus to start running the script only after waiting for 10 seconds to see if there is any AT command traffic on the serial port.
- Execute Python script
 - Once the start mode is set use the AT#ESCRIP="yourscriptname" command to execute the script and then restart the Terminus. The script will run after 10 seconds and it will run every time the Terminus is started.
 - To disable the script type in AT#ESCRIP="" during the 10 second interval after power up.

Other Useful Commands

AT#RSCRIPT - Read an uploaded script

AT#LSCRIPT - List saved Python scripts

AT#ESCRIP? - Shows the currently enabled script

9 Tracker Demo

9.1 Overview

The tracker demo is designed to give the user a fast way to bring the unit up for evaluation while also covering example code for various functions. The demo features the following for usage/evaluation:

- Automatic checks and settings for network registration and general operation
- File system creation and reading/writing
- Low power stand by event monitoring, wake up on event occurring.
- Real time event monitoring
- Automatic socket connection attempts for data reporting
- Store and forward data reporting during loss of network situations
- Remote control and status checks via SMS

9.2 Demo Package Contents

TTDemo.zip

- Python Scripts
 - ATC.py
 - exceptions.py
 - GPRS.py
 - JANUS_CONFIG.py
 - JANUS_GPS.py
 - JANUS_IO.py
 - JANUS_SER.py
 - JANUS_SMS.py
 - TTDemo.py
 - NETWORK.py
 - timers.py
- Netcat application
- Realterm application

9.3 Tracker Demo Checklist

- Personal Computer with Windows OS and internet connection
- RS-232 serial connection
- PythonWin IDE installed
- Power supply for the Terminus Tracker

9.4 Installing the Tracker Demo

Unzip the contents of the Tracker Demo archive to a local drive

9.5 Configuring the Demo

In order to evaluate the Tracker Demo, the customer must adjust the following parameters in the JANUS_CONFIG.py file.

```
#####  
##Begin User Entered Values  
#####
```

9 Tracker Demo continued

9.5 Configuring the Demo continued

BAND

Please refer to AT Command guide for AT#BND

If Terminal used in North America (BAND = '3')

DefaultConfig.BAND = '3'

NETWORK

If Terminal used on ATT / Cingular in North America (NETWORK = 'ATT')

Else (NETWORK = 'GSM')

DefaultConfig.NETWORK = 'ATT'

APN

Gateway Address for GPRS traffic

This setting is GSM Provider and possible customer specific when a VPN is used

This demo is defaulted with 'internet' that is used for ATT wireless settings from pre-Cingular days

You MUST obtain the APN setting for your GSM account. Please call GSM provider! Janus can't help you with this.

DefaultConfig.APN = 'userapn'

IP

IP address of server on the Internet which Terminus will connect to send and receive data.

Address in this example is not operational for customer evaluation. Customer must have their own server setup to interact with this demo.

IP = 'xxx.xxx.xxx.xxx'

DefaultConfig.IP = '12.237.120.180'

PORT

PORT number of server on the Internet which Terminus will connect to send and receive data

PORT number in this example is not operational for customer evaluation. Customer must have their own server setup to interact with this demo.

DefaultConfig.PORT = '7777'

PROTOCOL

If customer is using TCPIP (PROTOCOL = 'TCPIP')

Else leave blank (PROTOCOL = '')

DefaultConfig.PROTOCOL = 'UDP'

GPRS USER NAME

If GSM Provider requires GPRS User Name (USERNAME = 'JOE')

Else leave blank (USERNAME = '')

DefaultConfig.USERNAME = ''

GPRS PASSWORD

If GSM Provider requires GPRS Password (PASSWORD = 'JOE123')

Else leave blank (PASSWORD = '')

DefaultConfig.PASSWORD = ''

9 Tracker Demo continued

9.5 Configuring the Demo continued

SMS_ENABLED

SMS Enabled? (SMS_ENABLED = 'TRUE' or 'FALSE')

DefaultConfig.SMS_ENABLED = 'TRUE'

SMS

SMS Designation phone number (SMS = '+16305551212')

Although this number is put into the settings and checked for validity when pulled in from the configuration file, it's not used. For simplicity, the SMS functions of this demo take the incoming SMS and parse the originating phone number, replying to that phone number.

DefaultConfig.SMS = '+12345678910'

INTERVAL

How many seconds the demo will wait before sending new GPS data to server? Valid range is 1 to 86400

In this demo, GPS data will be sent every 10 seconds (INTERVAL = '10')

DefaultConfig.INTERVAL = '10'

Antenna LNA

Does the external antenna have an LNA that needs to be powered, meaning is it an active antenna (bias voltage supplied by the tracker)?

If the antenna is not an active type, enter 'FALSE', otherwise 'TRUE'

DefaultConfig.LNA = 'TRUE'

Switch Input

Are we utilizing the input N/O switch?

If we are going to use/monitor the switch, enter 'TRUE', otherwise 'FALSE'

DefaultConfig.NOSWITCH = 'TRUE'

Ignition Input

Are we utilizing the Ignition input?

If we are going to use/monitor the ignition, enter 'TRUE', otherwise 'FALSE'

DefaultConfig.IGNITION = 'TRUE'

Cellular Stat LED

Do we want it active, or not?

If we are going to have it active, enter 'ON', otherwise 'OFF'

DefaultConfig.SLED = 'ON'

GPS/User LED

Do we want it active, or not?

If we are going to have it active, enter 'ON', otherwise 'OFF'

DefaultConfig.ULED = 'ON'

9 Tracker Demo continued

9.5 Configuring the Demo continued

Auto-On Control

Do we want it active, or not?

If we are going to have it active, enter 'ON', otherwise 'OFF'

DefaultConfig.AUTOON = 'OFF'

Ignition Follow Control

This flag controls if we will be following the ignition signal for being on and reporting. If this is set, the unit will continue to report until the Ignition line becomes inactive. At that point, it will send a final report showing this, and then shut down. Note that this should only be used if AUTOON is OFF, otherwise the unit will just continue to automatically power up and power down if the ignition signal is inactive (car/truck is off).

This is useful for power savings also, since the unit will be idle until it's needed drawing minimal current from the battery.

If we are going to have it active, enter 'TRUE', otherwise 'FALSE'

DefaultConfig.IGNITIONFOLLOW = 'TRUE'

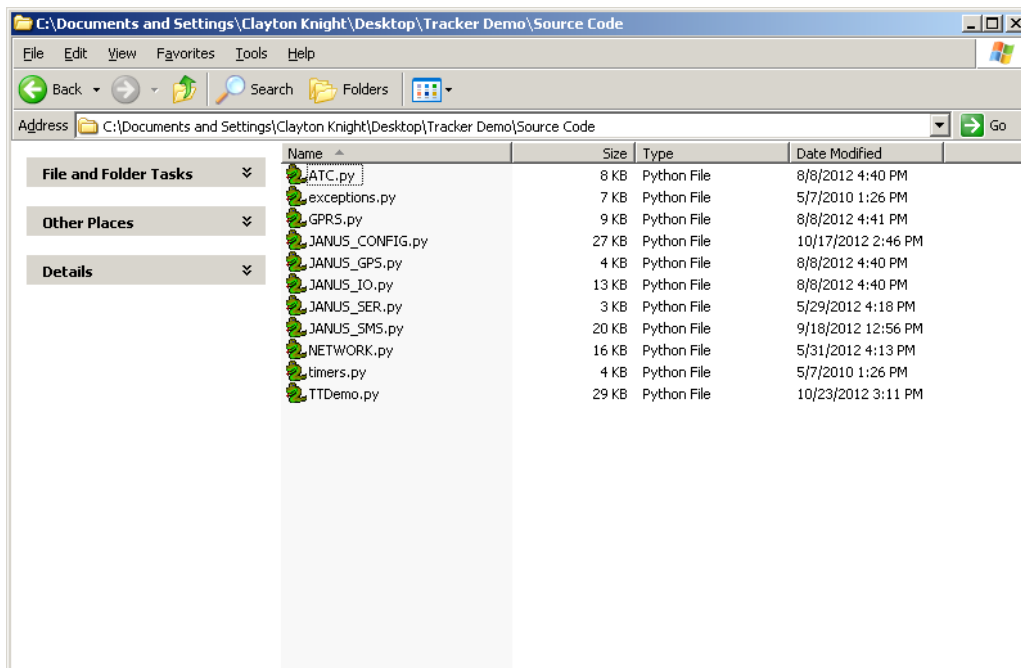
```
#####  
##End User Entered Values  
#####
```

Note that for the purposes of this demo, the below values are REQUIRED:

- DefaultConfig.IP = '12.237.120.180'
- DefaultConfig.PORT = '7777'
- DefaultConfig.PROTOCOL = 'UDP'

9.6 Uploading the Demo

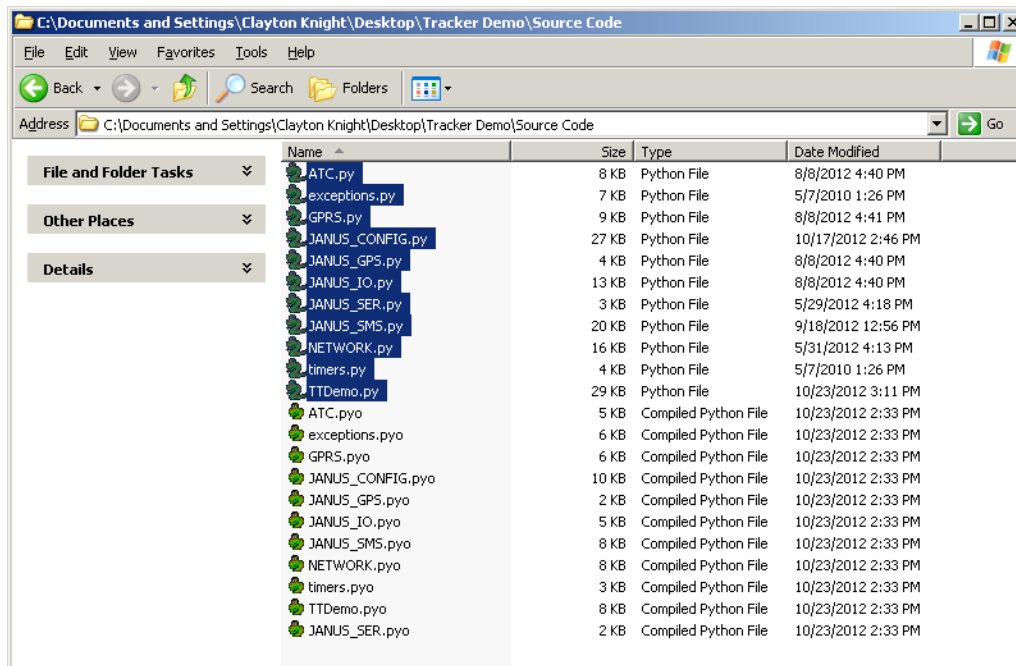
Open the directory that contains the Tracker Demo scripts with explorer.exe



9 Tracker Demo continued

9.6 Uploading the Demo continued

Right click and select “Compile”



All files should compile and additional files should exist with *.pyo extension.

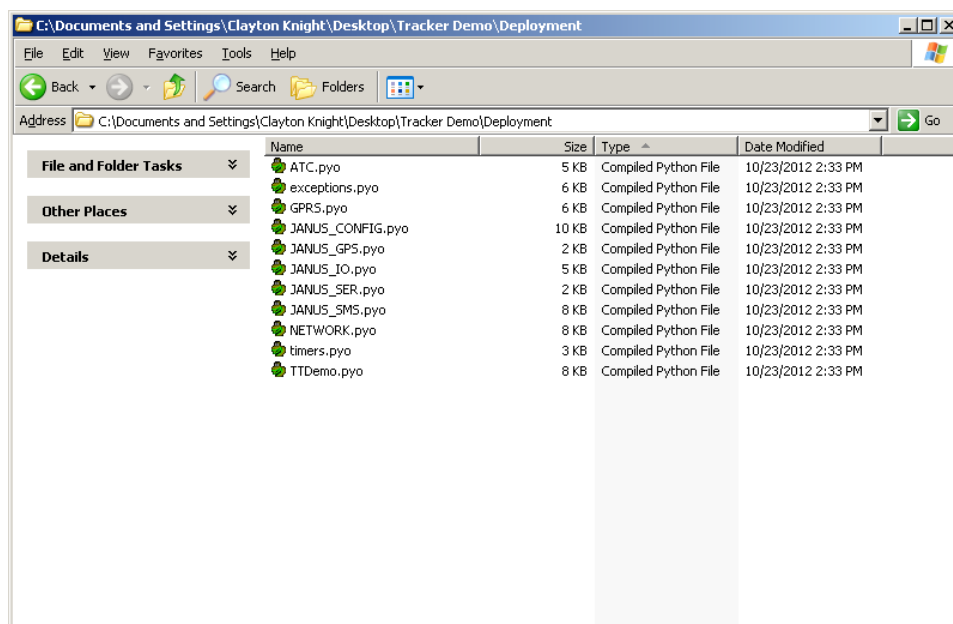
Move these files to the directory named Deployment.

Connect the serial port signals to the development computer serial port. The serial port of the development computer should be the serial port configured as the MDM interface in the Telit COM Port Selection Tool. Refer to the Terminus User Manual under Installing Pythonwin IDE for more information.

Ensure the GPS and Cellular antennas are connected to the Tracker.

Apply power to the Tracker, by default it will automatically start up once power is applied.

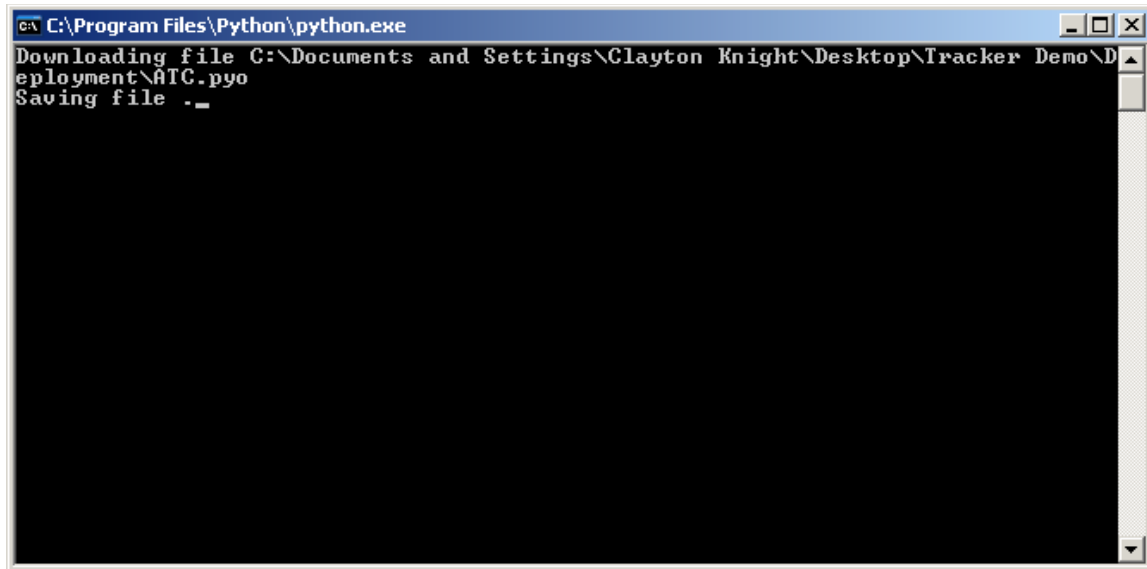
Open the Deployment directory that contains the Terminus GPS demonstration compiled scripts with explorer.exe.



9 Tracker Demo continued

9.6 Uploading the Demo continued

Right click download for each file in the Deployment directory.



9.7 Setting the script to auto start

Open Windows HyperTerminal and create a new connection. Select the serial port of the development computer that has been selected as the MDM interface in the Telit COM Port Selection Tool. Set as follows:

- Baud Rate: 115200bps
- Bits: 8
- Stop Bits: 1
- Parity: None
- Hardware Handshaking: Yes

Open the serial port and send the following commands.

- AT#ESCRIP=TTDemo.pyo<cr>
- AT#STARTMODESCR=1,10<cr>

9.8 Opening the Demo GUI

Open your internet browser and go to the address:

<http://gps.conwin.com/gpstrack2/>

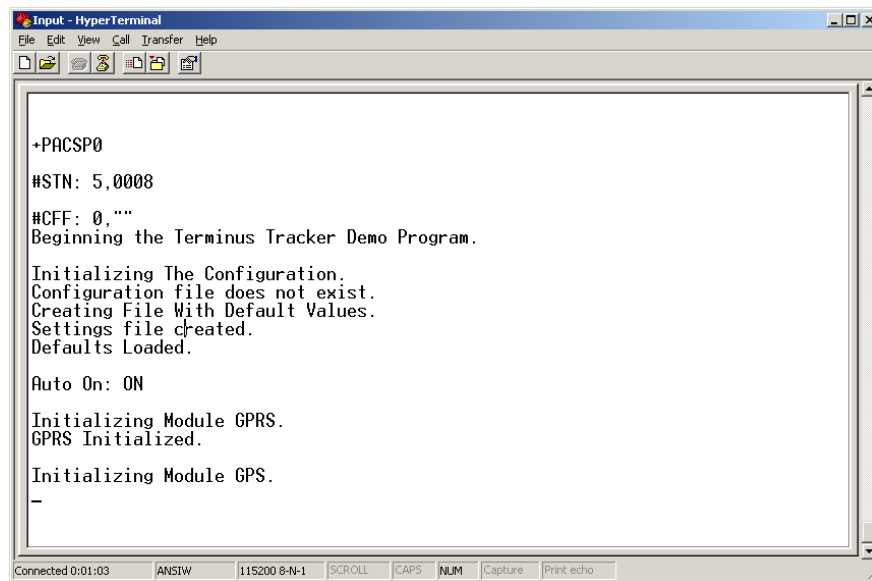
The left side will show all active units being tracked, and the main screen will give positioning on the map overlay.

9 Tracker Demo continued

9.9 Running the Demo

Power cycle the Tracker or simply enter AT#REBOOT

After approximately 10 seconds the demo script will start and data will appear on the terminal window.



```
Input - HyperTerminal
File Edit View Call Transfer Help

+PACSP0
#STN: 5,0008
#CFF: 0, ""
Beginning the Terminus Tracker Demo Program.

Initializing The Configuration.
Configuration file does not exist.
Creating File With Default Values.
Settings file created.
Defaults Loaded.

Auto On: ON

Initializing Module GPRS.
GPRS Initialized.

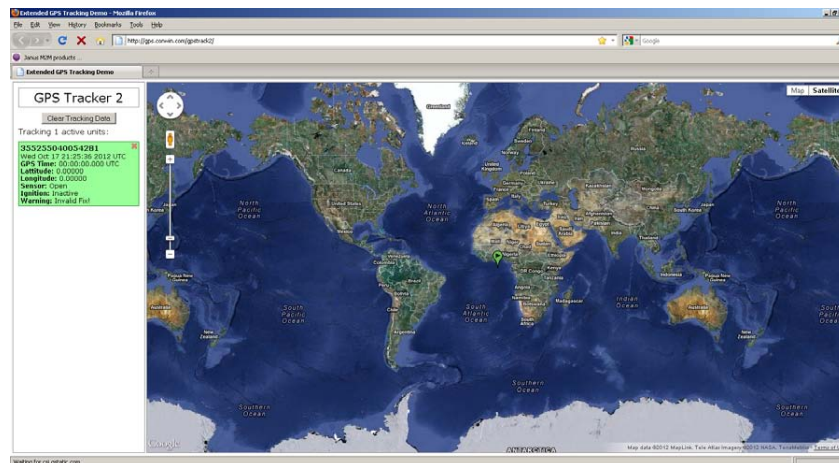
Initializing Module GPS.
-
```

On the first run through, the demo will take the configuration information entered previously into the script and write a configuration file. This file can then be adjusted on the fly by SMS. If the configuration file becomes corrupt or a value is unreadable for some reason the default value will then be used so it's important to set the default values to something safe.

Now go over to the Demo window in your internet browser, you should see your unit's ID with the data on the left side and an indicator in the main screen.

The information shown and sent contains the following:

- IMEI of the cellular modem
- The RMC NMEA data string
- Monitored data
 - N/O Open or Closed
 - Ignition Active or Inactive
 - Wake up Condition when Event Based operation is being used



The IMEI and NMEA information are required, but the other informatin can be turned off and will not be reported.

9 Tracker Demo continued

9.10 SMS on the fly configuration

The Demo supports receiving SMS commands to do 3 possible things:

1. Status query for the current reporting information and location
2. On the fly adjustment of how the unit will behave
3. Remote AT commands

The status query can be acquired by sending: **STATUS**

The demo will send an SMS back to the originating number with the following information:

- Unit: 1111111111111111
- Switch Reporting: TRUE
- Ignition Reporting: TRUE
- Status LED: ON
- User LED: ON
- Auto ON: ON
- Report Interval: 10
- Ignition Follow: TRUE
- Current Location: \$GPRMC,000000.000,V,,,,,,000000,,,N*4D

The tracker's reporting and function can be adjusted on the fly by a specific set of commands. Once adjusted, these settings are saved into memory by writing to a configuration file.

To change a function, send "CMD: x = y" (no quotation marks) where x is the command to change and y is the value. Below are the settings which are capable of being changed by this and what values the demo can accept:

- INTERVAL
 - Valid integer between 1 and 86400
- NOSWITCH
 - TRUE
 - FALSE
- IGNITION
 - TRUE
 - FALSE
- SLED
 - ON
 - OFF
- ULED
 - ON
 - OFF
- AUTOON
 - ON
 - OFF
- IGNITIONFOLLOW
 - TRUE
 - FALSE

Further explanation of these commands is in the "Configuring the Demo" section.

For example, to change the unit to turn Auto-On OFF you would send: **CMD: AUTOON = OFF**

The script will save the setting and respond to the originating phone number with an echo: **Auto ON : OFF**

The tracker can also accept and carry out standard AT commands. Once complete the unit will send back the response of the command.

For example, sending the basic "AT<cr>" command you would send: **AT** (No carriage return)

The script will save the setting and respond to the originating phone number with an echo: **OK**

Terminus Tracker Products User Manual

Front



Rear



Ordering Information

Ordering Information	Description
----------------------	-------------

Revision History

Revision	Revision Date	Note
A00	10/30/12	Advanced User Manual

JANUS REMOTE
COMMUNICATIONS

Division of The Connor-Winfield Corporation
2111 Comprehensive Drive • Aurora, Illinois 60505
630.499.2121 • Fax: 630.851.5040

www.janus-rc.com

Janus Remote Communications Europe

Bay 143
Shannon Industrial Estate
Shannon, Co. Clare, Ireland
Phone: +353 61 475 666