Application Note 102

August 2008

Authors:
David Kinsman, Program Mgr
Thomas Heck, Sr. Design Engineer
The Connor-Winfield Corporation

Implementing a Location Tracking Device Using the Terminus GSM864QP-V1.00

Location Tracking Terminal for Remote Communication Applications

Introduction

The Terminus GSM864QP is a wireless data terminal with versatile functionality in remote communications applications. With the current availability of compact, low-cost, low current cellular radio modules, GPS receivers and data-only services plans from GSM cellular radio carriers, location tracking using the Terminus GSM864QP is a highly sought after application.

The System

The basic blocks of the location tracking set up include; the tracking device, the cellular carrier infrastructure, the GPS system infrastructure, the Internet infrastructure, and the computer/server to query, receive, compile and display the user data. (see Figure 1)

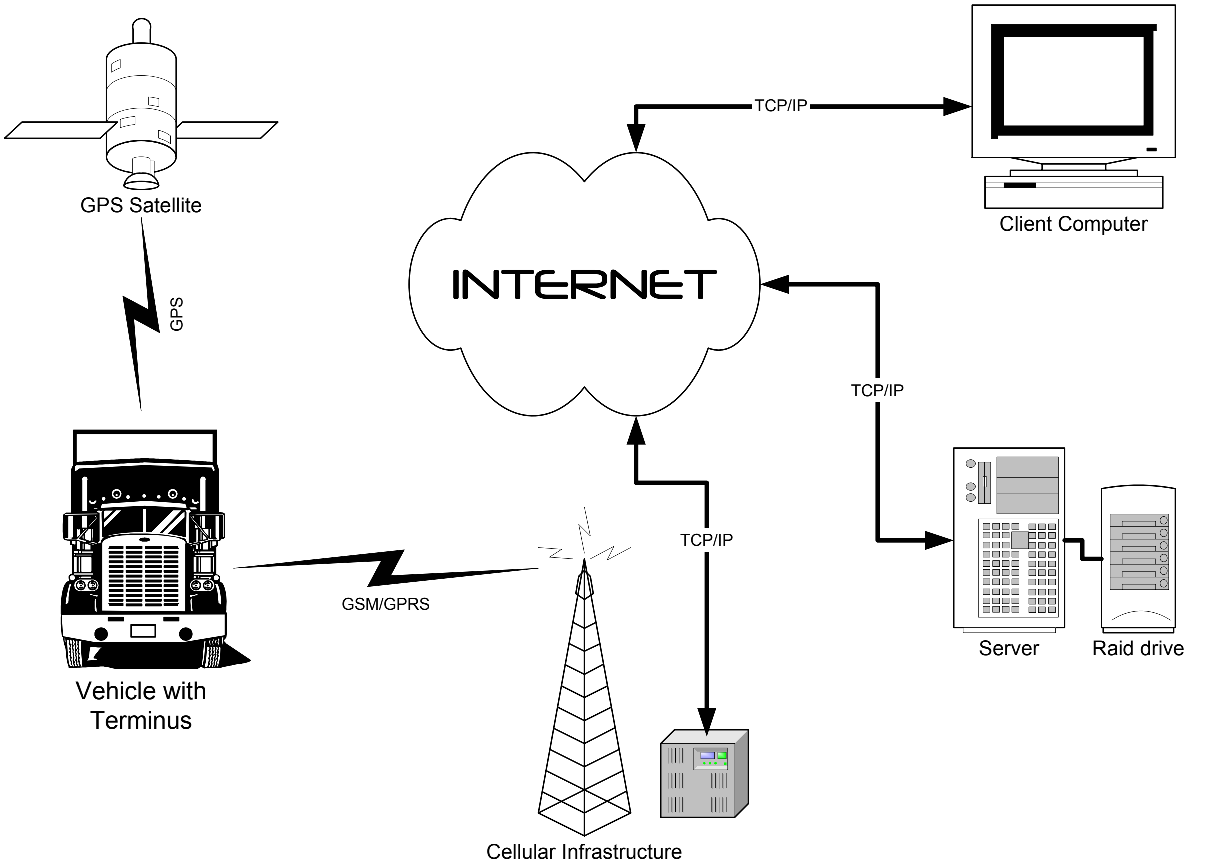


Figure 1

The Terminus

The Terminus is a wireless terminal with GPS capabilities that is powered by 7 to 18 VDC or by connecting directly to an automotive 12 V system. The Terminus contains the functional blocks detailed in Figure 2.

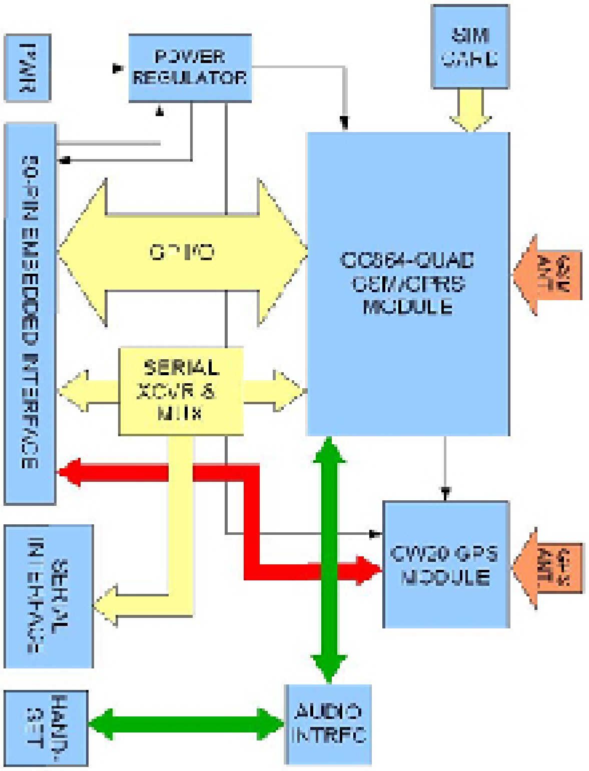


Figure 2

The Telit GC864QP GSM/GPRS Module – This provides the cellular radio interface that allows the Terminus to communicate via GSM Cellular Telephone Infrastructure. It also provides the processing engine that controls the cellular radio and script running capabilities for control and performance of various functions. Python scripting language is used for access to cellular radio functionality, 9 GPIO ports and the serial interface.

The SIM Card – Contains Carrier subscription and device specific information including messages and contacts.

The CW20 GPS Module – Receives the GPS satellite signal, extracts location & time data, and streams it out as serial data.

The Serial Transceiver and Multiplexer - Enables the Terminus to serially communicate with devices compatible to the RS-232 interface standard. In this application, the interface routes the serial GPS data to the internal Python application.

The Audio Interface – Enables user to send and receive audio information via the Terminus using a handset or other device. This interface is not used in this application.

Setting Up the Terminus (Please refer to the Terminus User Manual for details)

1. Obtain a SIM card from a Carrier
2. Add jumpers to the 50-pin Interface connector on the bottom of the Terminus connecting pins 43 and 45 together and pins 47 and 49 together. This routes the GPS data stream from the CW20 to the Telit Serial interface. Use 0.1” center “Berg” type jumpers.
3. Write the Python script
4. Compile and load Python script.

The Python Script

For this application, the Terminus device runs on a Python script that periodically initiates a GPRS session and sends out GPS NMEA data from the CW20 receiver corresponding to the current geographical position of the Terminus unit. The logic flow of this script is as follows:

1. Set up the operating parameters of the Terminus for interfacing with the selected GSM carrier
2. Power Terminus on and start the script. Once set up, the Terminus will register with the GSM network
3. Initialize GPRS mode in Telit module
4. Take control of the UART Select pin (GPIO 20) of the Telit module to switch data stream to GPS NMEA data
5. Initialize CW20 GPS receiver and send out data
6. Connect to the tracking server over the Internet and upload GPS data
   a. Set GPRS parameters: IP Address, IP port, socket number, user name and user password
   b. Open a socket
   c. Upload GPS data
   d. Close socket
   e. Repeat as often as desired for tracking resolution

The Python script used for this demonstration is included for reference in the appendix.

The Tracking Server

The tracking server is the user interface component of the system. It consists of a Server PC and a Client PC, both connected to the Internet. Details are outlined in Figure 3.

Server Computer

1. Listener – Monitors the port connected to the IP address receiving GPS data from the Terminus. This interface routes incoming GPS data to the Data Store
2. Data Store – file where the GPS data from the Terminus is stored
3. CGI – Common Gateway Interface – services requests from the Client for GPS data and returns data from the Data Store
4. HTTP Server – services HTTP request from clients and returns data when requested

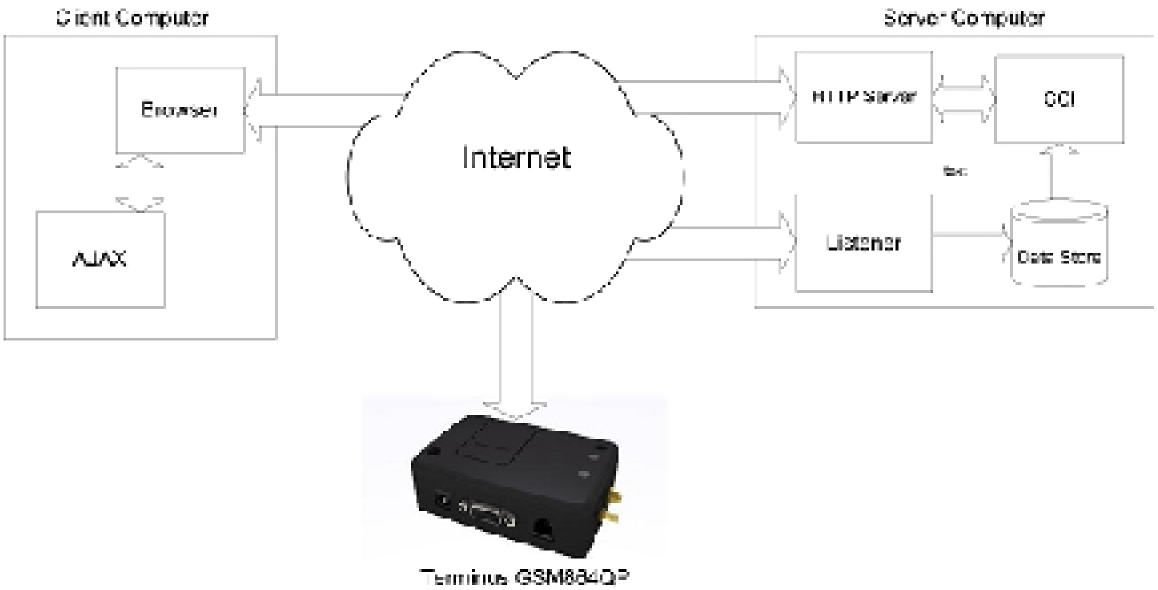
Client Computer

1. Browser – in this case Firefox, which access Google Maps for position display.
2. Google Maps API Browser Plug-in (not shown)

The Tracking Server allows the user to access tracking information on any computer with Internet access. When running, the client computer displays the Terminus' location superimposed onto the Google Maps page as a moving object.

Conclusion

The location tracking application detailed in these application notes is a fairly basic example of the Terminus GSM864QP's capabilities as a mobile telemetry device. This system can be enhanced by having the Terminus monitor various status indicators in a vehicle and by using the audio interface for emergency communications. Additionally, these enhancements can be monitored on the client computer.



Appendix

The Python scripts included as examples are as follows:

- **ATC.py** – Defines various classes (functions) used by the main program
- **CW20.py** – External GPS Class Example
- **exceptions.py** – Defines exception handling hierarchy for Python scripts
- **locationdemo.py** – Main routine that configures the Terminus, sets up the GPRS call, sends out status SMS messages and send GPS NMEA data to the Server
- **timers.py** – Defines various timers used by the program

For a complete listing of the Python scripts used for this application, please contact Janus customer support at golp@janus-rc.com.

Authors: David Kinsman, Program Mgr
Thomas Heck, Design Engineer
The Connor-Winfield Corporation