

# Low cost PDA/Gps based field logging solution for GRASS data

Alessandro Frigeri\*, Giorgio Minelli\*

\* Dipartimento di Scienze della Terra, Geologia Strutturale e Geofisica, Università degli Studi di Perugia,  
P.zza Università 1, 06100 Perugia, ITALY, tel. ++39 0755852623, Fax ++39 0755852603,  
e-mail afrigeri@unipg.it

## Abstract

The need of electronic data to be inserted in geospatial databases lead us to develop a simple Personal Digital Assistant (PDA) based field log system to store punctual data to be imported directly in GRASS as a site-file. The application communicates with a GPS unit so positional data are automatically stored, allowing the user to input only categorical data. The system frame has been chosen so that the overall cost of a basic system is very low but no changes are required to use it with professional GPS units and different PDA systems. The software is released under the terms of GPL license so that everyone could improve it or adapt it to his own field logging campaign.

## 1 Introduction

Current technology allows us to bring computational and digital storage systems on the field. Specialised applications however are not so economically favourable to be produced in large scale so that cost of particular systems may be very high. By utilising some end-user products we can reach good level of automation in field logging activity.

PalmGSites/PalmGeoSurvey (PGS) is an application that allows to use a low-cost Personal Digital Assistant (PDA) together with a common GPS receiver to store data in the field and then transfer them into a GRASS database.

The developing platform has been chosen keeping in mind two things:

- even scientists with no deep experience in programming could adapt the software to his own needs.
- the code should be portable to different devices and should be able to run also on cheaper low-end PDAs.

The first item implies that the code is available and the latter that we must find some common run time environment that should run on the most diffuse PDAs.

## 2 Why Free Software?

Free software is a software whose source code is shared openly to the public. 'Free' does not mean 'at no cost'. 'Free' does not even mean of public domain. A public domain software could be used for any reason by anybody but the authors has got complete control on it, so that he could decide to 'close' the code and make it unavailable for any reason. With Free Software copyright is still claimed by authors that use it to keep access to the code open to everyone [1]. Free software uses the General Public License (GPL) to protect the availability of code and prevents that code will not be included in proprietary binary-only system.

Benefits coming from the use of free software are several. The main evidence is that institutions and individuals that do not possess financial resources for commercial software could go ahead with their research activity without worrying about licenses fees, concentrating financial resources in acquiring scientific skill. The core benefit stands in that science should be based in sharing knowledge and not in hiding it.

So a code could be developed by all the people interested in it, with different programming skills and different point of views. This style gives birth to a huge number of projects, and GRASS is one of them.

Another benefit is that academic codes, e.g. carried out during a PhD, will not be lost as PhD student finishes his studies. In this way even complex projects could grow in the years.

### 3 PGS logging system

The software presented here is called PGS, where P stands for palm computer, G stands either for GPS GRASS or Geo , and S stands for Survey or Sites since that the system is made to log GRASS site data.

**Developing Language** Since the commercially available PDA systems architecture differs slightly, we choose to develop the application with WABA (<http://www.wabasoft.com>). WABA is a free (under GPL license) developing environment for PDAs that uses a Java dialect as programming interface. Compiled source can then run on different PDAs on top of WABA virtual machine to be installed together with the application. Since now, WABA has been ported to Motorola Dragonball based devices as PalmOS and Hand-spring devices, TI calculators, Sharp Zaurus and Compaq iPaq running linux. That seemed a good deal since Motorola Dragonball devices could cost less than a half than Intel StrongArm based platforms.

**Developing platform** PalmGSites has been developed on a Debian GNU/Linux system and tested on PalmOS IIIxe PDA system (cheapest solution), the GPS unit is a low-end Garmin eTrex receiver. The data transfer between PDA and PC is made through pilot-link <sup>1</sup>, a set of tools to communicate with a PalmOS PDA over a serial port. The conversion between pdb format (used by all PalmOS applications) and GRASS is made with a Python module called PeepDB written by Alexey Vyskubov <sup>2</sup>.

**GPS-PDA Communications** Every GPS unit has got its own proprietary communication protocol, but all the devices have got the option to output data in NMEA format. To maintain usability of software with different GPS unit the latter format has been used as it represents a standard in navigational systems communications.

### 4 The NMEA protocol

The National Marine Electronics Association (NMEA) is a non-profit association composed of manufacturers, distributors, dealers, educational institutions, and others interested in peripheral marine electronics occupations. The NMEA 0183 protocol was first released in March of 1983 and defines the electric signals requirements, data transmission protocol, timing and specific sentence formats for a EIA-422 serial data bus. EIA-422 serial bus could be adapted to the RS232 interface simply by connecting the single wire TTL line of EIA-422 to RS232 receive pin. The NMEA 0183 settings for a RS232 interface are a baud rate of 4800bps, 8 data bits, no parity and 1 stop bit. The electrical interface standard allows a single talker and several listeners on a line.

The GPS information is sent by means of ASCII sentences. Every sentence starts with a \$ character and ends with a <CR><LF> sequence and could not exceed 80 characters. Data fields are separated by commas and the absence of data will leave an empty field between two commas.

In table 1 a typical NMEA 0183 (version 2) sentence is reported. After the starting character the two letters GP indicate that a GPS receiver is transmitting (NMEA supports several different navigational devices). RMC is the ID that identifies the type of information which follows in the sentence, in this case data fields report the recommended minimum specific GPS/Transit data.

The NMEA protocol provides also query sentences to obtain specific information from a device (GPS receiver or other navigational systems).

In the case of Garmin eTrex GPS receiver data is one way from the device.

### 5 PGS

PGS setup is presented in figure 1. PDA is connected to GPS via a serial cable with a null modem adapter so that data transmitted by the GPS are received by the hand held computer. GPS communications have to be set so that NMEA protocol will be used.

The software consists in a database application with the functionality of open serial communication and grab NMEA information from the PDA's serial port. The Graphical User Interface (GUI) showed in figure 2 is simple and presents data fields to be filled

<sup>1</sup>pilot-link is available at <http://www.pilot-link.org>

<sup>2</sup>PeepDB is available under GPL license at <http://linux.piter-press.ru/peepdb/>

\$GPRMC,192456,A,4306.601,N,01225.974,E,00.8,0.0,030702,1.7,E,S*22<CR><LF>	
\$	Start of sentence
GPRMC	GPS receiver sending RMC sentence
192456	Universal Time Coordinate (UTC): 19:24:56z
A	Navigation receiver: A=OK, V=Warning
4306.601	Latitude 43 deg. 06.601 min
N	Latitude North
01225.974	Longitude 12 deg. 25.974 min
E	Longitude East
10.8	Speed in knots
0.0	Course in degrees, true North
030702	Date: 3rd of July, 2002
1.7	Magnetic variation, degrees
E	Magnetic variation, East
S*22	Checksum

Table 1: NMEA 0183 v.2 RMC sentence

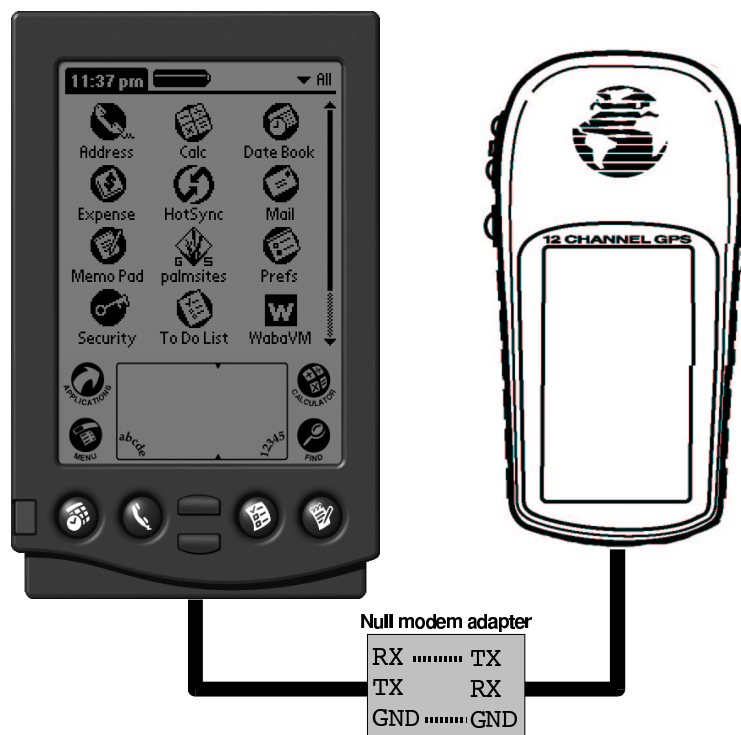


Figure 1: Setup of PGS system



Figure 2: PGS: The GUI

in a structural geology survey. WABA together with Waba Extras extension written by Rob Nielsen <sup>3</sup> allows to insert either simple data field to be filled, or pup-up lists with selectable entries, as the case of feature entry in figure 2.

Once the information has been entered, user can choose to read the position from gps receiver tapping on the GPS button. The position could be also entered directly by hand. If the site data is good, tapping on the save button saves the data and a new empty page to be filled is presented. Previously acquired data can be browsed with arrows button.

Once the survey is finished the user can transfer data into GRASS connecting the GPS to the computer with GRASS.

Dataset can be transferred with pilot-xfer program from the pilot-link suite (that has to be previously installed in your system).

The pdb file has to be converted into an ASCII GRASS-site file to be imported into GRASS with s.in.ascii. The conversion is made with pdb2gasites.py, a simple Python script (available in PGS distribution) that uses PeepDB module by Alexey Vyskubov.

## 6 Conclusions

The PGS system allowed us to develop a working data logging system without large expenses. This will facilitate geologic field work and speed up data storage in electronic format. PGS runs on a wide range of PDAs and also on cheaper low-end ones, so that also private experimenters or students could deal with it.

The choice of Free Software opens the code to the public so that eventually present bugs (not recognised at the moment) can be fixed, and the software usability improved. The PGS code is available through the Internet at <http://www.unipg.it/~afrigeri/PGS>.

## 7 Future development

As free software, anyone can contribute to it. Feedback is welcome and will be really precious to improve and extend the functionalities of the software. It would be interesting to hear reports of PGS on different platforms and GPS devices. Since the WABA language is very simple, even non computer scientists could study it and adapt it to their own needs.

Future development could include a software module to customise data fields directly on the PDA, a synchronizing application to load data into GRASS in one shot and all the features that people would like to have in PGS.

<sup>3</sup>Source code is available at <http://www.cygnus.uwa.edu.au/~rnielsen/wextras/>

---

3Com, PalmOS, Compaq, Sharp, Texas Instruments, Garmin, Motorola, Intel are registered trademarks of their respective holders. The devices used in this article are privately own by the authors and thereafter they do not have any direct relation with relative companies.

## References

- [1] Jr. John W. Stockwell. The CWP/SU: Seismic Un\*x package. *Computer & Geosciences*, (25):515–419, 1999.