HP50GPS: A GPS Solution for the HP50g Calculator (version 1.0)

By: Pepin Torres, P.E. (KB1JIB)

Abstract:

HP50GPS is a set of programs designed to capture and display NMEA 2.2-complaint GPS data on the HP50g calculator[*]. It can extract, display, and save position and tracking data as well as capture and save satellite information data. The programs are written in UserRPL and SysRPL.

[*] While I have never used an HP49g+ or an HP49gII before, I see no real argument against using it on these two models provided the right set of hardware is used.

Raison d'être:

I wrote these programs to give back to the HP calculator community and to add to the available catalog of HP↔GPS serial interface software. Programs like Above 2.0 (which served as inspiration) provide limited GPS capabilities and I wanted to contribute a more comprehensive and expandable set of programs that can be useful to owners of serial GPS units, especially those without any type of GUI. I own two Deluo GPS units both without any interface except for a serial connector. When not in use for APRS purposes (and other Amateur Radio stuff), these units are basically mini-bricks. My intention was to have a portable interface that can make these units even more useful by expanding their capabilities.

System Requirements:

1. HP50G (should work also with HP49g+ and HP49gII)
2. Serial data cable (available on the commerce site of hpcalc.org)
3. Serial GPS unit that is NMEA 2.2-complaint (assume 4800 baud, 8 data bits, 1 stop bit, no parity and that GGA, RMC, GSA and GSV sentences are sent.)
4. SD Card (optional)

Installation:

Copy all the files into any HOME subdirectory of your choice.
The application starts by pressing the `GPS` soft menu key.

This software was written and tested using:

- HP50G
- Serial cable ordered from hpcalc.org
- Deluo GPS (http://www.deluogps.com/product.php?id=30-504-01). This GPS must be powered from an external 5V source.
- HPUserEdit 4 and the bundled emulator.
- Conn4x (on a PC) and Link48 (on a Mac SE)

UserRPL Software Included:

- `GPS` : Main program. This will display a CHOOSE window to select a desired mode of operation.

- `POSIT` : Captures positional data from the GGA sentence and displays it in an INFORM box.

- TRACK : Captures tracking data (speed/course) from RMC sentence and displays it in an INFORM box.

- ACTIV : Captures data on the active satellites used by the GPS. Version 1.0 puts the GSA raw sentence as a string on the stack and returns to main window.

- VISI : Captures data on the active satellites visible by the GPS. Version 1.0 puts one GSV raw sentence as a string on the stack and returns to main window.

- LOGGR : Captures all the raw sentences (GGA, RMC, and GSA) every 5 seconds until 50 sentences are captured (roughly five minutes) and then dumps them into a file in the SD card (port :3:). The log file is automatically named using the current time on the calculator (example: a file named LG1027 means log file was saved at 10:27). LOGGR is quasi-infinite loop that can only be stopped gracefully by pressing CANCEL (ON key) at which point the current collected data is dumped to a file in the SD card and returns control to the main executable (GPS). The files on the SD card can then be either read back by the user to work with or transferred to a computer. This type of calculator file may require a better text editor than notepad.exe or wordpad.exe to view and manipulate. By the nature of the NMEA sentence, the output file is a comma-separated value file and thus easily manipulated with any spreadsheet software.

- HOWFAR : Captures current location (GGA sentence) and determines the heading and distance to a coordinate input by the user. The program also allows selecting from a set of pre-defined points. This feature is not available in version 1.0.

- CHKSUM : This supplemental program calculates the checksum for the given GPS sentence and compares it with the checksum sent (for NMEA, the checksum is the 2 hex numbers after the asterisk.) This program expects the raw sentence as a string on the stack. Returns 1 if checksum passed, 0 otherwise.

SysRPL Software Included:

- GGA, RMC, GSA, GSV : Compiled SysRPL programs that read the raw sentences from the serial GPS. If successful, they put the sentence on the stack as a string on level 2 and the number 1.0 on level 1. If failed, it returns 0. These were written in SysRPL because speed was a factor and because the author wanted to work on SysRPL for a change!

- GGA.u, RMC.u, GSA.u and GSV.u : SysRPL source code for the programs above. Need to have Library 256 installed on the calculator to edit and compile.

Caveats:

1. The **GPS** program puts the calculator in the correct state before running and resets it to the default state upon exit. (Default state: USB, 115K baud. Run-time state: Serial, 4800 baud). The individual programs can be tweaked for standalone operation by changing the source code, but it is suggested they follow the code in GPS to put the calculator in the right state before and after running (this will save a lot of key presses.)

2. The process of scanning serial streams, manipulating, parsing and doing checksums on NMEA sentences is quite the chore for the calculator. Expect delays of 3 to 5 seconds between start of capture and display of data. Also, a lot of effort was made in incorporating error trapping code to catch any runtime error which may surface (a disconnected serial cable, GPS unit powered off, GPS unit coming out of startup, etc.)

Serial Port Buffer Limitations:

The HP50G serial port buffer is 255 bytes long. This means that for a typical GPS unit that puts out the GGA, RMC, GSA and GSV sentences every second, the buffer will fill up very quickly indeed. Note that each sentence can be a maximum of 82 or so ASCII characters, and a typical GPS will put out 1 GGA, 1 RMC, 1 GSA and 1 or more GSV sentences every second. The GSV sentence is expected to have multiple parts by design. Assuming 3 GSV sentences (about 12 satellites' worth of information), this means that in 1 second the GPS will put out 5 sentences at 82 characters per sentence = 410 characters. This implies that even when the buffer is empty, after one second of operation the input buffer will be full and it may or may not have the complete sentence the program is looking for. Below is a typical 1 second output from a Deluo GPS:

```
$GPGGA,045841,4323.5933,N,07114.4670,W,1,05,01.5,-00048.3,M,050.8,M,,*7F
$GPGSA,A,3,09,15,18,21,22,,,,,,,,02.8,01.5,02.3*0A
$GPGSV,3,1,12,05,07,147,30,06,03,272,00,09,67,097,42,12,09,131,26*74
$GPGSV,3,2,12,14,24,257,27,15,24,064,42,18,83,019,45,19,05,313,00*7C
$GPGSV,3,3,12,21,42,197,46,22,49,306,46,24,10,166,32,26,05,064,27*79
$GPRMC,045841,A,4323.5933,N,07114.4670,W,000.0,178.4,111108,,,A*61
```

This data set is 403 characters long. If the buffer was empty at start of collection, only this would be available after 1 second:

```
$GPGGA,045841,4323.5933,N,07114.4670,W,1,05,01.5,-00048.3,M,050.8,M,,*7F
$GPGSA,A,3,09,15,18,21,22,,,,,,,,02.8,01.5,02.3*0A
$GPGSV,3,1,12,05,07,147,30,06,03,272,00,09,67,097,42,12,09,131,26*74
$GPGSV,3,2,12,14,24,257,27,15,24,064,42,18,83,019,45,19,05,~~313,00*7C~~
~~$GPGSV,3,3,12,21,42,197,46,22,49,306,46,24,10,166,32,26,05,064,27*79~~
~~$GPRMC,045841,A,4323.5933,N,07114.4670,W,000.0,178.4,111108,,,A*61~~
```

Since the capture can start at any point in time (i.e., before, during or after the 1 second burst is sent), there is a very high probability that the desired sentence will be truncated. For this reason, the programs that read from the serial port buffer (the SysRPL programs GGA, RMC, GSA and GSC) require that the input buffer be cleared before every attempt at collection. This is done via the UserRPL commands CLOSEIO and OPENIO run consecutively. Please note that when reading a sentence from the serial port, the code will enter a loop in which the calculator will:

1. Clear the input buffer.
2. Wait at least 1.0 seconds.
3. Read 1 character at a time until "`$`" is found. This is the start-of-sentence delimiter.
4. Read 6 characters and compare them to "*`XXXXX,`*" where that string can be "`GPGGA,`" , "`GPRMC,`" , "`GPGSA,`" or "`GPGSV,`" depending on the sentence desired.
5. If matching, read 76 characters and concatenate the previous 6 characters to form an 82 character string. Return the string and a 1.0 to the stack. Note that 82 is the maximum sentence length.
6. If not matching, go to step 1.
7. If the reads (done with the SysRPL command DOSRECV) take longer than 10 seconds or so, a timeout error will occur and control given back to the main program.

Applications:

As a ham radio enthusiast, I use my calculator and HP50GPS as part of an APRS tracker that I have installed in my car. By putting a home-made serial port splitter between the Deluo GPS and a TinyTrak3+ unit, my calculator can listen in to the NMEA strings being sent to the APRS tracker. Thus, I am both sending position packets over the air on 2 meters and capturing them on my SD card. This works out very well because the TinyTrak3+ provides +5V power to the GPS so I don't need to deal with a power supply. Please note that the TinyTrak3+ provides power on pins 4 and 5 (5V and ground, respectively.) Putting a straight splitter may cause damage to something in the chain; this is the reason for using a home-made splitter. So be careful when trying to interface the calculator to devices that provide power.

To learn more about APRS see http://en.wikipedia.org/wiki/Automatic_Packet_Reporting_System
To learn more about the TinyTrak3+ go to http://www.byonics.com

And finally a note from the author:

I had a lot of fun writing this software and even more fun making it work. As with all works of art, it may never be finished, only abandoned. Hopefully, I won't be abandoning this project before version 2.0 comes out. In the meanwhile, use it and let me know how it works for you. If you find any bugs or would like new features incorporated, drop me a line. You may email me at pepin (dot) torres [at] gmail (dot) com.