DAPHNE: A CLIENT/SERVER DATA ACQUISITION AND DISTRIBUTION SOFTWARE. APPLICATION WITH LOAD CELLS, GPS POSITIONING AND ACCURATE EVENT TIMESTAMPING ON OCEANOGRAPHIC SHIPS

Version 1.1.0

Giuseppe Stanghellini
Giovanni Bortoluzzi

DAPHNE: A CLIENT/SERVER DATA ACQUISITION AND DISTRIBUTION SOFTWARE. APPLICATION WITH LOAD CELLS, GPS POSITIONING AND ACCURATE EVENT TIMESTAMPING ON OCEANOGRAPHIC SHIPS

by Giuseppe Stanghellini and Giovanni Bortoluzzi

Includes bibliographical references and index.

1. Data acquisition 2. Client/server 3. Oceanographic ships

Abstract - We present DAPHNE, a client/server Java software that acquires data and serve to CLIENTS on the Internet. The software was tested on Load Cell tranducers (cable layout strength) and GPS (NMEA, TSIP) on board of R/V OGS-Explora. A particular application deals with the accurate timestamp of events, by using the Acutime^{TM}GPS receivers, especially useful for refraction seismic.

Sommario - Viene presentato DAPHNE, un software in modalita' client/server per la acquisizione di dati e la loro distribuzione su Internet. Il software e' stato testato con Celle di carico (tensione di cavi) e GPS (NMEA,TSIP) a bordo della nave oceanografica OGS-Explora. Una applicazione particolare riguarda la registrazione di eventi ad alta accuratezza temporale, tramite il ricevitore GPS Acutime^{TM}.

ACRONYMS

ACRONYM	DESCRIPTION	URL-email
CNR	Consiglio Nazionale Delle Ricerche	www.cnr.it
ISMAR-Bologna	Istituto di Scienze Marine CNR Bologna	www.bo.ismar.cnr.it
INOGS	Istituto Nazionale di Oceanografia e Geofisica Sperimentale	www.inogs.it
NMEA	Nat.Marine Electronic Association	www.nmea.org
CLIENT/SERVER	Client/server Application	client/server
JAVA	SUN	java.sun.com
RXTX	RXTX	www.rxtx.org
FUGRO	STARFIX	starfix
ASHTECH	AQUARIUS	aquarius
ACUTIME	Trimble	acutime pdf
TSIP	Trimble Standard Interface Prot.	tsip
AEP	AEP	www.aep.it

Table 1: Acronyms of Organizations, Manufacturers, Products

AUTHORSHIP

Giuseppe Stanghellini (g.stanghellini@bo.ismar.cnr.it) wrote the software. He is a technical and scientific software programmer, with particular skills in system programming and interfacing (mainly with C/C++ and Java compilers). Giovanni Bortoluzzi (g.bortoluzzi@bo.ismar.cnr.it) installed and tested the system.

GETTING DAPHNE

The software can be obtained from ISMAR-CNR. Please contact the authors.

DISCLAIMER

The software is provided 'as-is' and is not redistributable. The present version is considered an alpha release. ISMAR is not responsible in any way for damage to equipment or personnel due to the use of this software. Use it at your own risk.

ACKNOWLEDGEMENTS

We wish to thank Dr. Giuliano Brancolini of INOGS who gave us the opportunity to test and use the software onboard the R/V OGS-Explora, and to M.Grossi and the technical staff of the ship for help in the installation. The use of Acutime^{TM} for the highly accurate event timestamping was encouraged by Dr. B.DellaVedova of the University of Trieste. Mr. G. Lioi and Mr. Cattini of AEP helped and assisted us while interfacing the AEP Load Cells. We thank also Dr. D.Jones of ISOF CNR for reviewing the manuscript.

Contents

List of Figures

List of Tables

1   MATERIALS AND METHODS

Throughout the text, the terms server and client may have a context-related meaning. According to [1], when dealing with the software context, "the client instantiates the user interface of an application that resides on a local or remote server". In contrast, when dealing with the hardware context, the server is the computer which handles the physical connection to the sensors, and the client is the computer that instantiates the connection to the server, possibly being the same server computer. To avoid confusion, we will quote in lowercase the software `server` and `client` applications, whereas we will quote in uppercase the hardware `SERVER` and `CLIENT` computers.

Overview

Daphne is a client/server software that collects data from several RS-232 inputs, stores the information in ASCII files with the proper timestamp and positioning data through a GPS port, when available, and serves them on demand to local `clients` or `CLIENTS` on the Internet. The `CLIENTS` may have the possibility to store data locally whenever required. The system is configurable through a GUI, for both the `server` and the `client` .

The software is based on single-thread (`server` side) and multi-thread (`client` side) programming models, and allows the following:

- ease of extension of instrumental capabilities

- minimization of resources required for running the tasks

- simplification of instrumental port management

- decoupling of the client GUI from raw data acquisition

The software is written in the highly portable Java^{TM} programming language.

Description

Our initial goal was the acquisition of Load Cell Data (cable strength) for dredging, coring and sampling purposes, positioning, accurate timestamping of events and other geophyisical data on an oceanographic ship.

The positioning data were supposed to be provided through a standard NMEA GPS interface, in particular the well known GGA, GLL and ZDA sentences, or a combination of them, or, at least, the GGA. With a minimum overhead, the ZDA sentence can be used to synchronize the computer clock to the UTC time of the GPS clock. This can be useful when no NTP `servers` are available on the network. A Trimble TSIP interface was also made available, particularly for the Acutime^{TM} receiver. Amongst many others instrumental interfaces we may cite for example magnetometers, gravity meters, cable layouts, and so on. The GUI interface was designed to be flexible enough to accommodate all the necessary parameters for each single instrument. Once the `server` application is launched on a particular TCP/IP port, one `client` or many `CLIENTS` are allowed to connect and visualize and store the ongoing data. The software can have both the server (option –server) and the client (option –client) active, however, the `server` instance

should be used only by trained personnel, since it is the core of system and it can be prone to errors or problems if not properly programmed. The client side is much less prone to errors and can be freely distributed to clients. A more powerful, fully featured software instance, allows the concurrent presence on the same GUI of the `server` and `client` runs.

The acquired data are available in a user selected directory as set during the initialization phase. They can be copied on the net using the standard Unix tools. The data arriving from each sensor are recorded 'as-is' on disk. The (sort) merge of the data in individual files is relatively easy since each record have the timestamp. The unix time field can be appropiately used to achieve this. See Appendices C containing data and some pieces of code for reading it.

Work in progress

- cable layout interface

- `client` plotting and presentation of navigation data

- system, GUI and port interface generalization

Planned improvements

- Data acquisition over TCP/IP network ports.

Assumptions and Requirements

The software runs on GNU/linux boxes (`server` side), and GNU/Linux, WIN32 and other O.S. environments (`client` side). A fully functional Java^{TM} environment is required for software development, such as the SUN's Netbeans IDE platform. Standard-compliant run-time Java^{TM} machines and WWW Browsers are required for data acquisition and visualization. The distribution comes with the RXTX software suite for RS-232 serial and parallel port control. A Java^{TM} archive file (RXTXcomm.jar) already tested on GNU/Linux and WIN32 boxes is provided. This can be run from the command-line, or placed in a directory accessed by the CLASS_PATH environment variable, or put in the standard directories for library extensions in the specific Java run-time environment.

The software must be run as the root user, to avoid problems in the RS-232 port environment.

2 USE OF THE SOFTWARE

This section presents the installation and use of the software. In particular, we will focus on the installation onboard R/V OGS-Explora. We had the opportunity to produce the software for the Antarctic cruises of Summer 2003-2004 of the PNRA (the Italian National Antarctic Project). OGS-Explora is a multi-purpose research vessel owned and operated by INOGS (Trieste), and was chosen to run seismic (reflection/refraction), morphobathymetric (multibeam), magnetometric and gravimetric data acquisition as well as sampling (coring/dredging).

2.1 Load Cell and GPS

The A-frame on the stern of R/V OGS-Explora have several deployment and suspension points. Among them, the central one was served by an AEP 20 T Load Cell, whose analog data were digitized by the AEP digital transmitter model TA5 and placed on an RS-232 line. Fig.1 shows the Load Cell. The GPS data were provided by either a Fugro STARFIX DGPS or a ASHTECH Aquarius GPS receivers, with the GGA (and ZDA) sentences. Fig.2 shows the system interfacing, and Appendix A show the electrical connections from the Load Cell/Transmitter to the `SERVER`.

Figure 1: The T20 AEP load cell on the OGS-Explora A-frame.

Figure 2: Block-diagram of the installation onboard of R/V OGS-Explora.

Software setting and usage

We present hereafter the –full instance of the software, which can be run only on the `server` . The other options –server and –client allow just single runs of the `server` and `client` . The snapshots that follow illustrate the setting and the feature of the program.

In particular, Fig.3 to Fig.11 show the setting and start of the `server` , including the choice of Load Cell and GPSNmea sensors.

The `server` and `client` instances have their own preferences:

- general, to set for example the TCP/IP port number and the verbosity

- I/O, to select the data directory and base names

- sensors, to create and connect sensors.

Each particular sensor have their own preferences:

- general, to provide general setting, such as the data acquisition rate

- device, to control the port communication parameters

- other, to deal with specific sensor controls and settings, such as zeroing or calibrating the Load Cell.
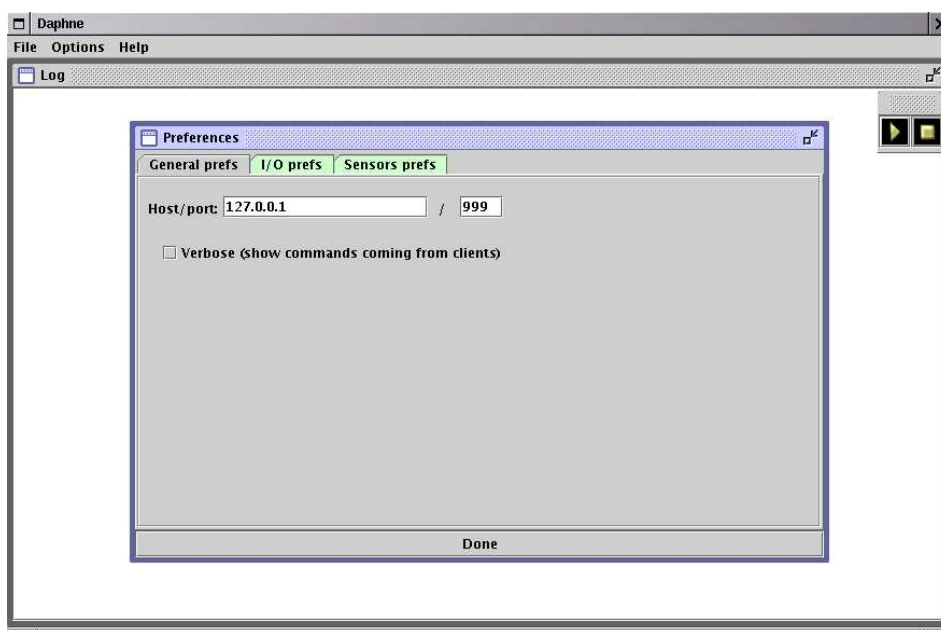
Figure 3: Server setting, general preferences. The IP number can be left untouched.
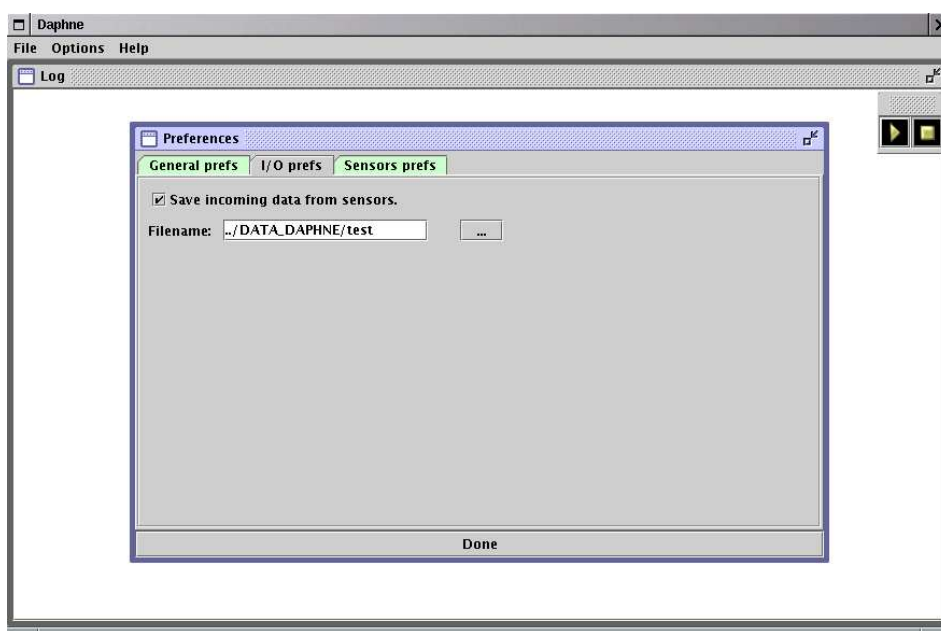
Figure 4: Server setting, I/O preferences. The user can select the directory and basename of the acquired files. The name of the files is then constructed by adding the sensor name and an hourly timestamp.
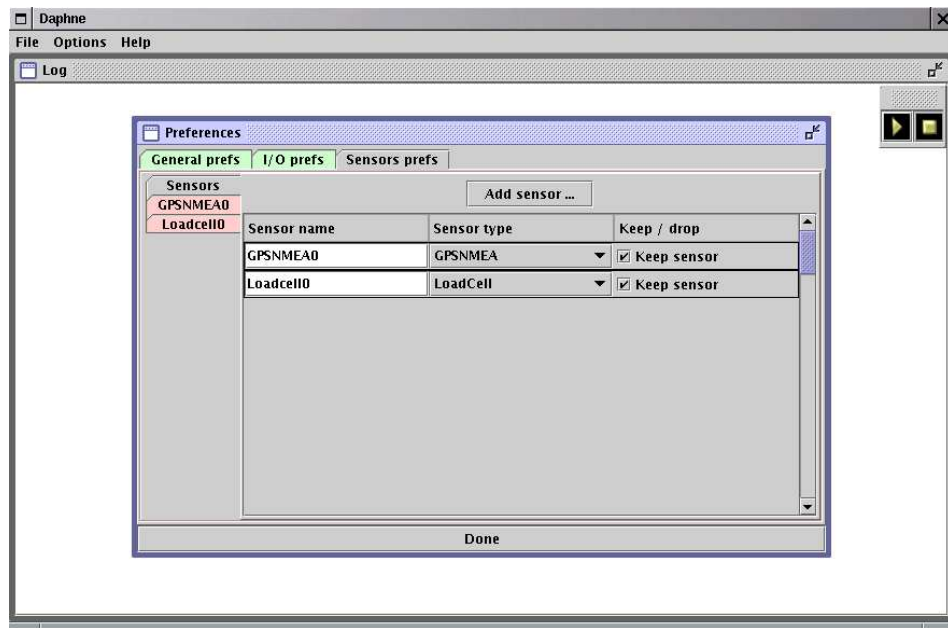
Figure 5: Server setting, sensors. LoadCell and GPSNMEA sensors are added to the system.

Figure 6: Server setting, Load Cell, general preferences.

Figure 7: Server setting, Load Cell, Other preferences. Note the Unit of Measure, zero and calibrate buttons.

Figure 8: Server setting, GPSNmea, general preferences.

Figure 9: Server setting, GPSNmea, Device preferences.
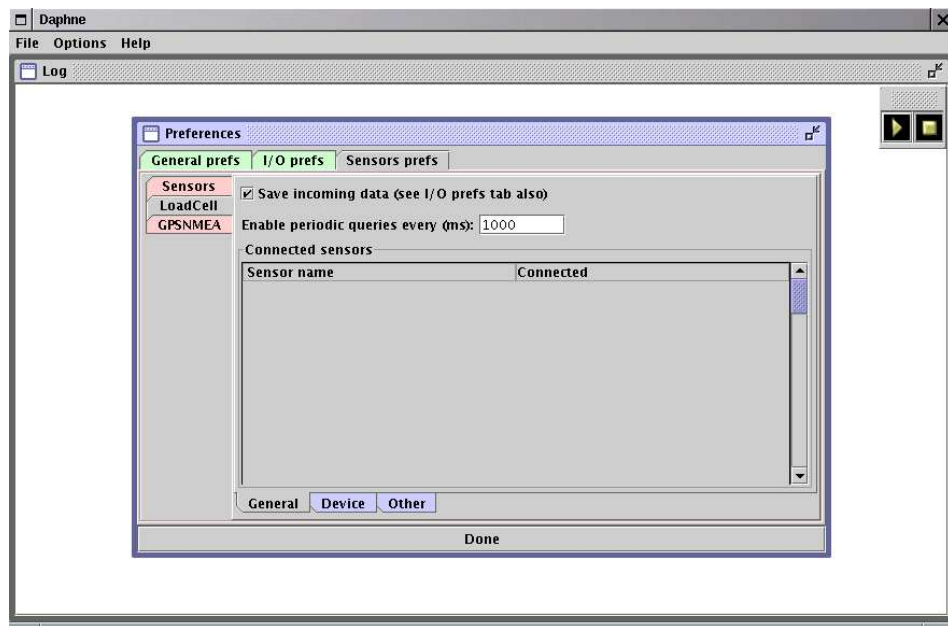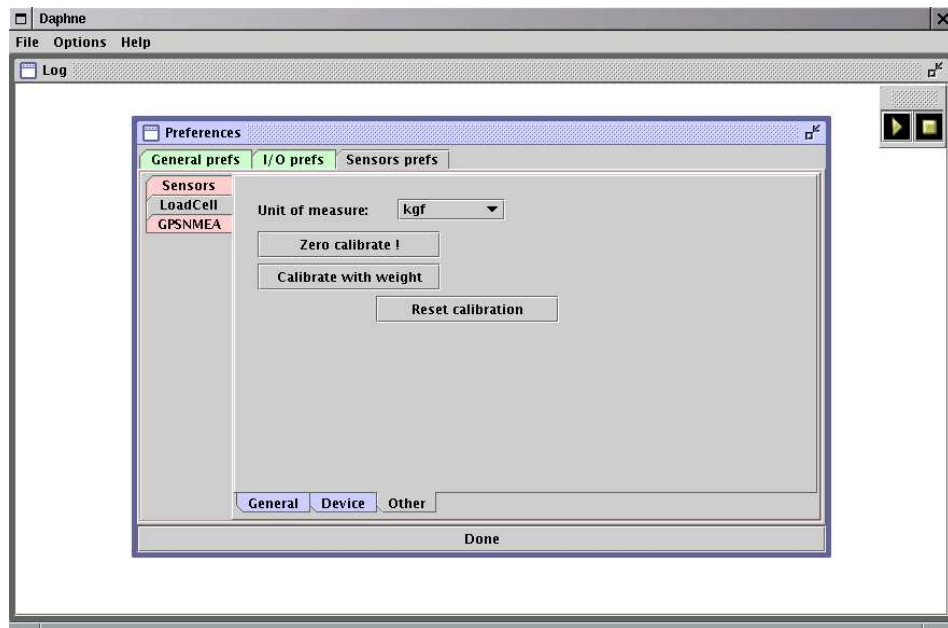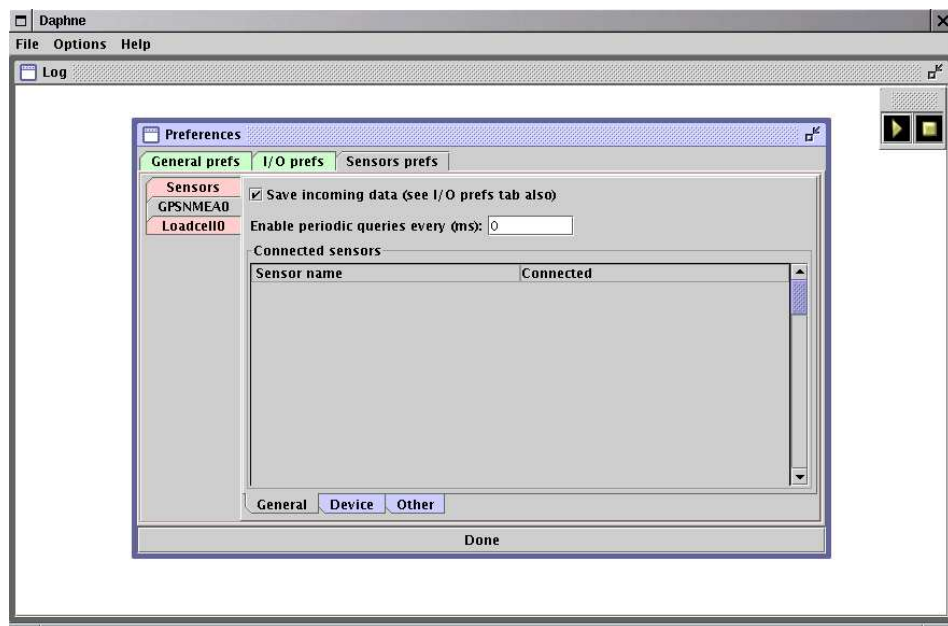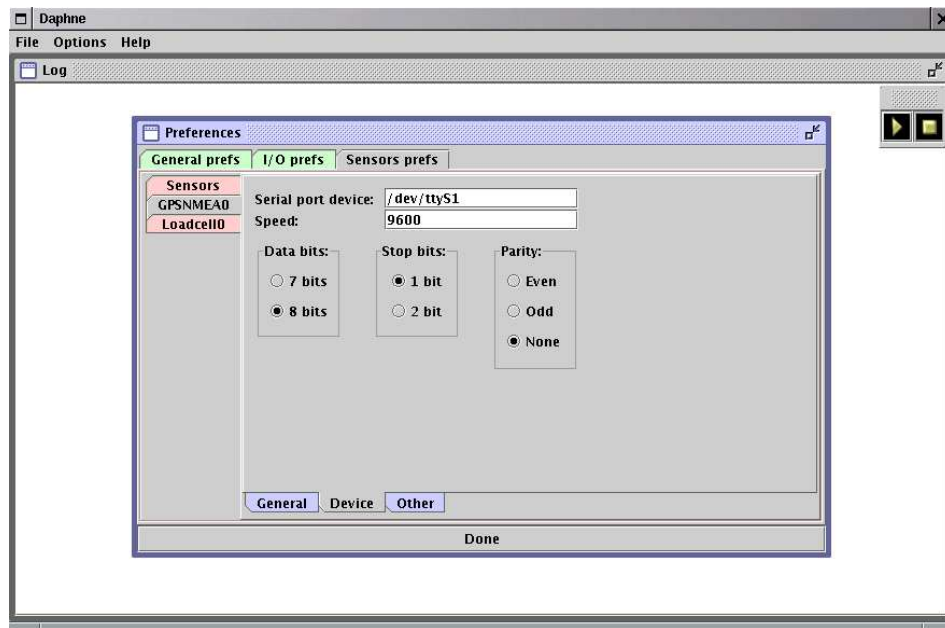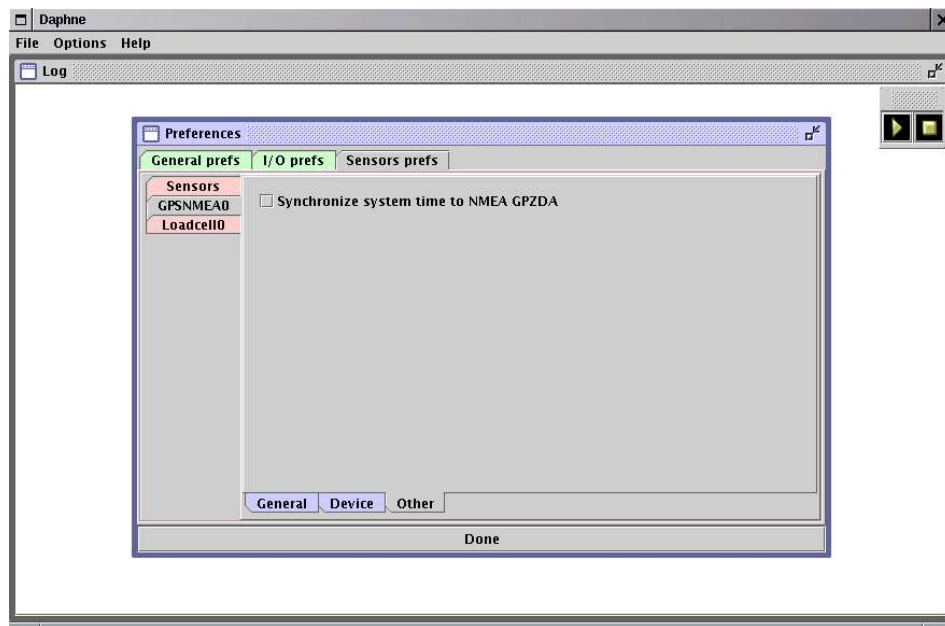

Figure 10: Server setting, GPSNmea, Other preferences. Note the time (UTC) synchronization button.
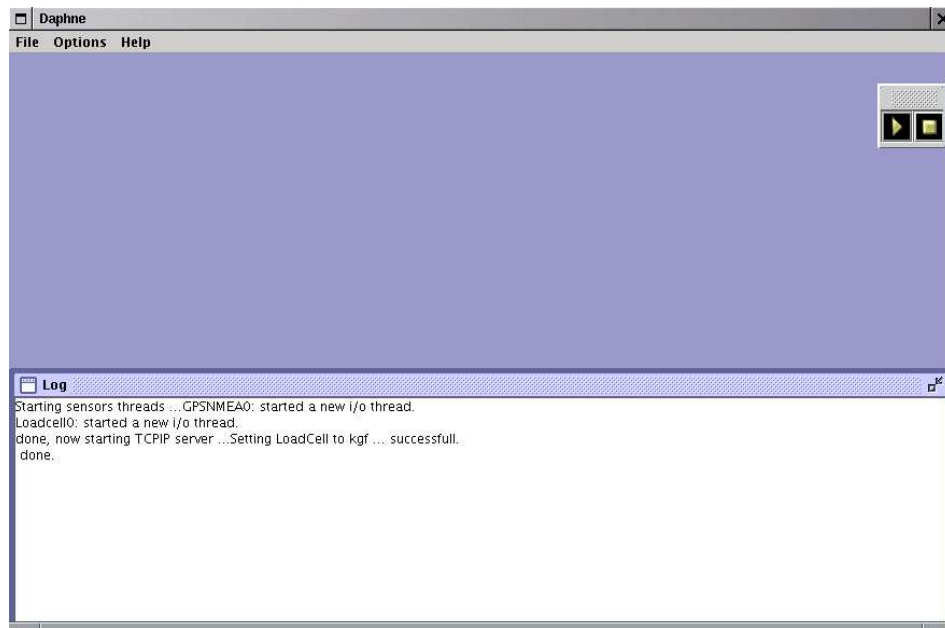
Figure 11: Server instance, start.

When all is done the user should press the <go> button (the well-known triangle icon above right in the window). To stop, the square button is pressed. The data are then acquired in the selected directory with an hourly timestamp for each individual file (this to prevent accidental file closure or destruction). The data have the UTC and unix timestamp as well, to make sorting and merging easier. See in Appendix C some example of the data acquired.

Once the `server` was launched and started to collect and store data, the `client` (Fig.12) can be used to visualize the incoming data (Fig.13).

The client instance is run from either the `SERVER` machine than remote ones. It shares the same behaviour of the `server` , with some limitations. Once the general preferences and i/o preferences are set, the <go> button is pressed to establish the connection with the `server` . This latter sends the proper information about the sensors it is connected to, and the client sets the main window accordingly (see Fig.14, that shows the graph of the Load Cell vs. time).

The `client` instance does not allow to select/create sensors, yet relaying on the `server` information.
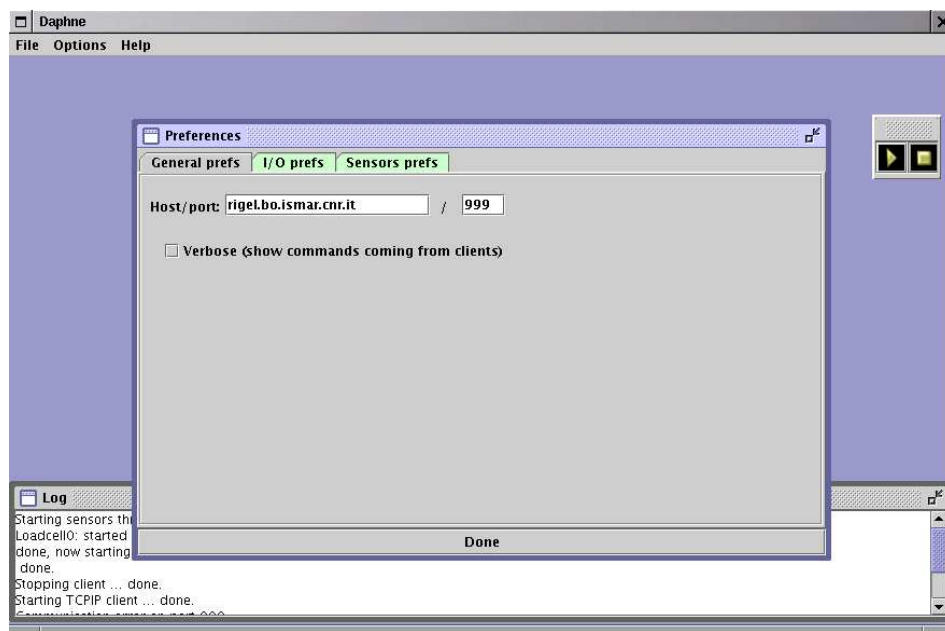
Figure 12: Client setting, general preferences. The user must set the IP address or domain name of the SERVER, and the port number of the server.
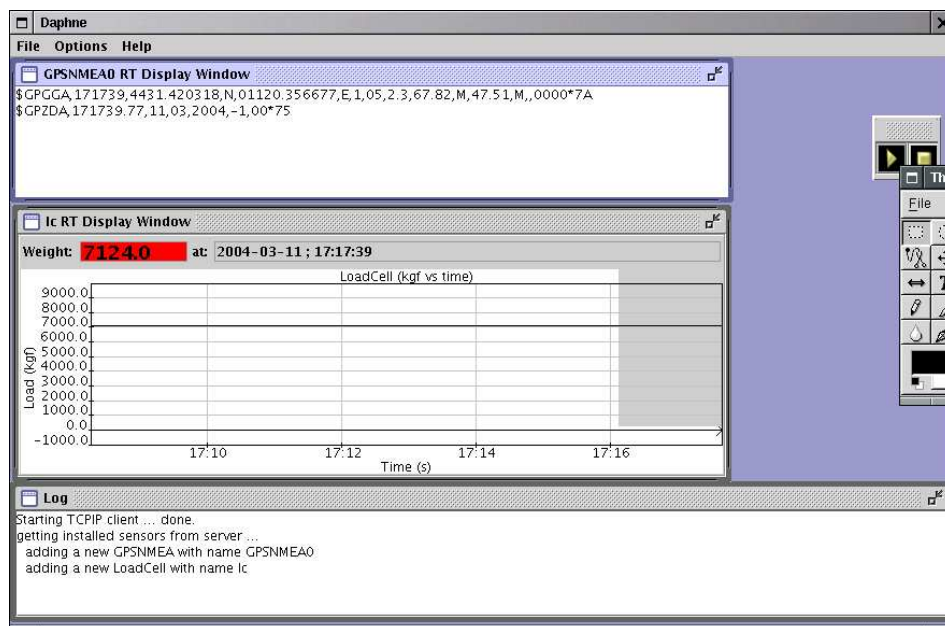
Figure 13: The client visualizing data.
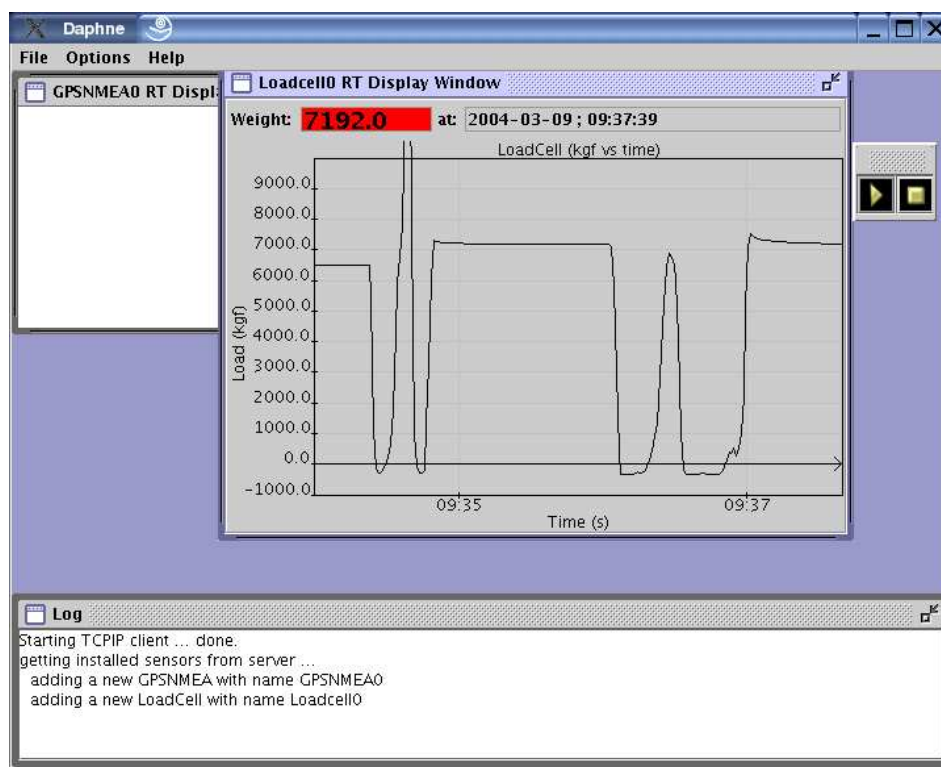
Figure 14: Client instance, Load Cell with graph.The red value turns to green when lower than the warning value set.

2.2 The Trimble Acutime^{TM}accurate event timestamping

Another class was created to interface devices that run the Trimble TSIP Protocol. In particular we interfaced the Trimble Acutime^{TM}2000 GPS (A2K) receiver and programmed in order to capture and timestamp events such as seismic shots or similar. This task requires accuracies of at least +/- 1ms or better. The A2K system is capable of giving accuracies of few tenths of nanoseconds, by using an overdetermined solution to obtain better time measurements from the highest possible number of SV available.

Figures 15, 16 and 17 show the panels for the instrumental setting. To fill the initialization strings user must press the buttons on the rigth of the white board. The control data bytes are reported in Appendix B. The event timestamp is triggered by pulsing the RTS line (pin 4-7) of the RS-232 port A. The port B is used for the GPS setting. See in Fig.18 the cable roouting and interfacing.

Figure 15: Acutime^{TM} event setting, general.
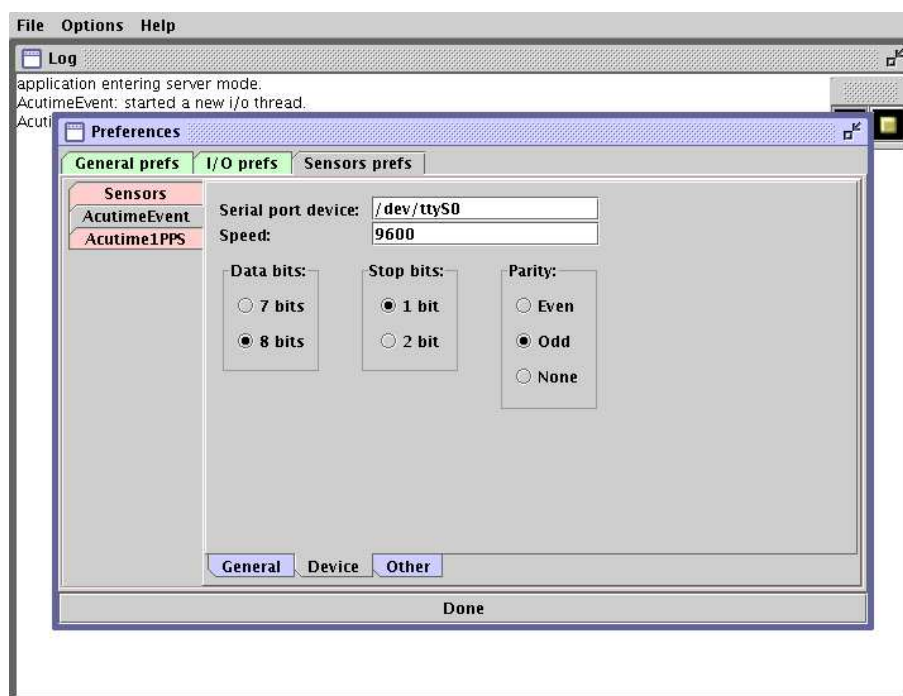
Figure 16: Acutime^{TM} event setting, device. Note also the odd parity setting.
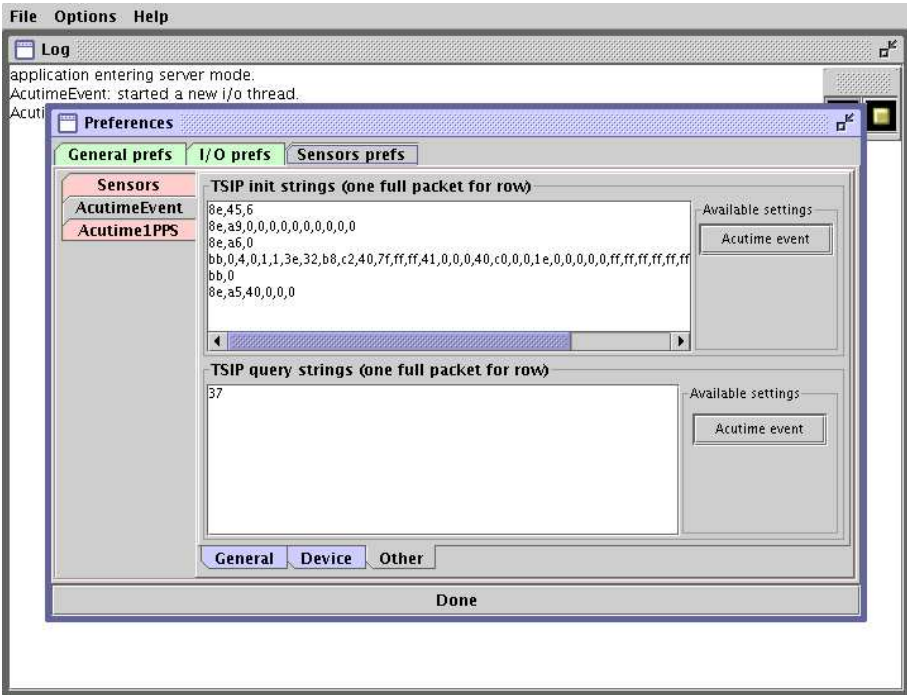
Figure 17: Acutime^{TM} event setting, other.
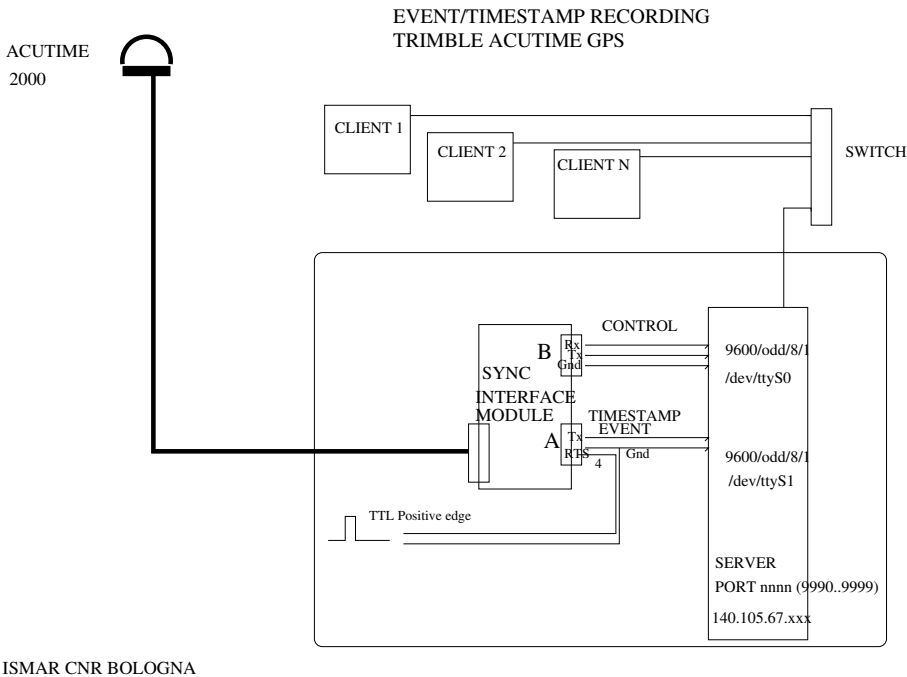
Figure 18: EVENT/TIMESTAMP cable routing and interfacing.

2.2.1 Tests and Results

This section presents here the results obtained while testing the event timestamping with the Acutime[TM]
interface. We connected the event input of the Acutime[TM] Synchronization Kit to a Quartz-Controlled Time Base Generator, set to 59.2 seconds. The time differences between each timestamp were then plotted on a Time Series plot, as shown in Fig.19. At a first glance we are able to see that the maximum error is about 250 microsec, to be taken into account for seismic purposes. The signal rise side started at 2004-02-11 09:00, there was lack of data 16:30 to 17:30 and a steep descending step began at 18:00. This behaviour should be investigated further, either on the GPS constellation management or on the TB side. In addition, there were some spikes, both for positive or negative time differences. In spite of the above problems, we can see that the data seem to be quite stable, and well within the Acutime[TM] specification.
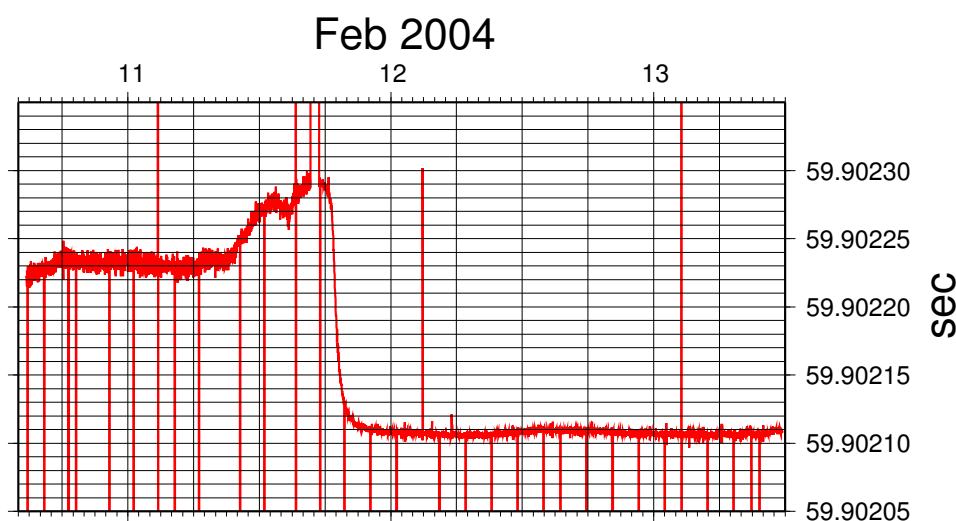
Figure 19: EVENT/TIMESTAMP. The time differences between events are shown.

3   CONCLUSIONS

The software we have presented, yet in a preliminary stage, is thought to perform reliably in data acquisition and distribution. Due to its modular concept and straightful connection to the Internet, it is supposed to be useful for applications requiring remote control or acquisition, other than local ones. The already available World Wide Area Network topology, and the emerging Satellite and WiFi Communication links give the possibility of a direct connection to mobiles. We have presented here an application on board of oceanographic ships. Even if not tested, a `CLIENT` on shore may connect to the `server` on board, once the Internet connection was available and established.

References

[1] Wikipedia, the free Encyclopedia (2001) www.wikipedia.org

[2] AEP (2003) Manuale d'uso del trasmettitore TA5. AEP Report. www.aep.it.

[3] Acutime 2000 Synchronization kit (2003) Trimble, Ca, USA. acutime pdf.

A   Connection of the AEP Load Cell

Fig.20 shows the electrical connections of the T20 Load Cell to the TA5 Digital RS-232 Transmitter. The system was installed with a data cable length of aproximately 45 m (from the stern A-frame to the recording room. The system worked fine with a power supply of 12.5V. However, 24 V should be preferred if available.
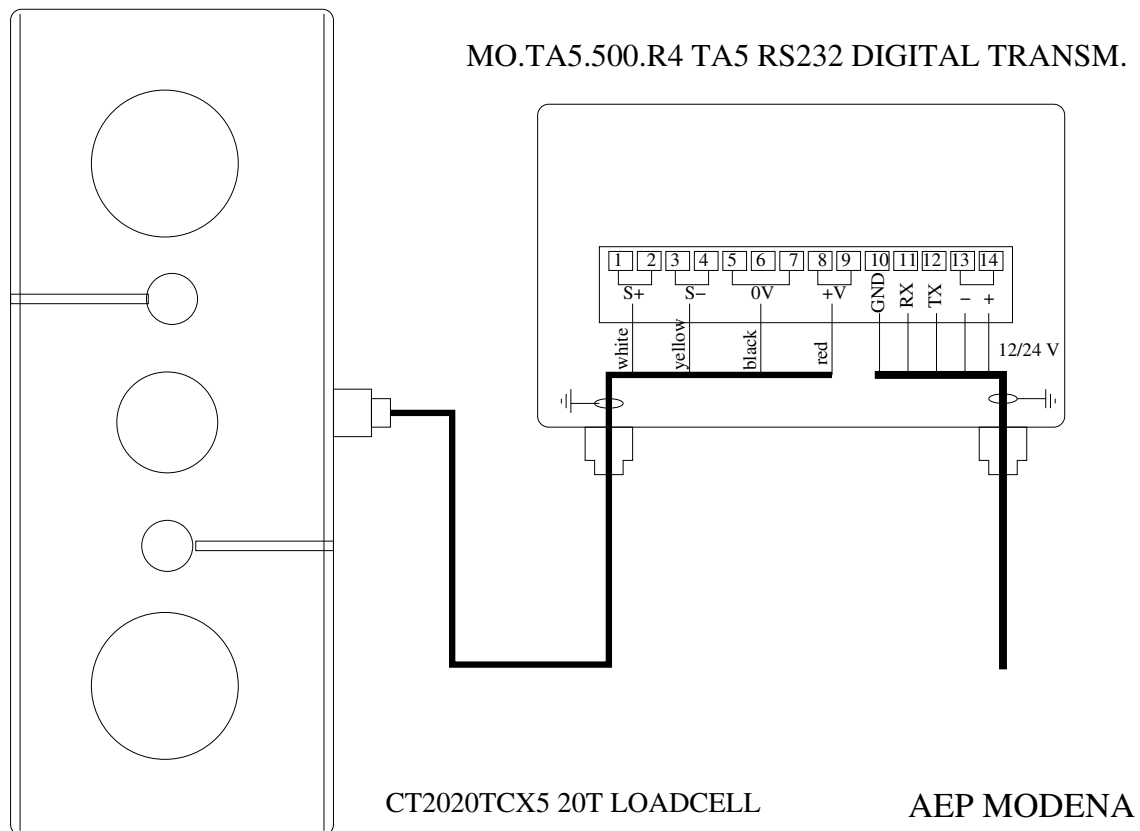
Figure 20: Electrical connections of AEP's T20 Load cella to TA5 transmitter.

B   Setting of the Acutime$^{\text{TM}}$

The A2K requires a proper initialization (according to the TSIP binary protocol):

- `8e,45,6`select accurate positioning

- `8e,a9,0`disable self survey `8e,a6,0` disable self survey

- `bb,0,4,0,1,3e,32`init and set the receiver `bb,0` init and set the receiver

- `8e,a5,40,0,0,0` event and `8f-ad`packet on port B

C examples of data

Hereafter the listings of the data directory is shown. Note the hourly timestamped sensor data files.

```
-rw-r--r--  1 root    root     323874 Mar 10 19:59 test.2004-03-10-18.GPSNMEA0
-rw-r--r--  1 root    root     266761 Mar 10 19:59 test.2004-03-10-18.Loadcell0
-rw-r--r--  1 root    root     410400 Mar 10 20:59 test.2004-03-10-19.GPSNMEA0
-rw-r--r--  1 root    root     338118 Mar 10 20:59 test.2004-03-10-19.Loadcell0
-rw-r--r--  1 root    root     410363 Mar 10 21:59 test.2004-03-10-20.GPSNMEA0
-rw-r--r--  1 root    root     338212 Mar 10 21:59 test.2004-03-10-20.Loadcell0
-rw-r--r--  1 root    root     410400 Mar 10 22:59 test.2004-03-10-21.GPSNMEA0
-rw-r--r--  1 root    root     338118 Mar 10 22:59 test.2004-03-10-21.Loadcell0
-rw-r--r--  1 root    root     410437 Mar 10 23:59 test.2004-03-10-22.GPSNMEA0
-rw-r--r--  1 root    root     338118 Mar 10 23:59 test.2004-03-10-22.Loadcell0
-rw-r--r--  1 root    root     410400 Mar 11 00:59 test.2004-03-10-23.GPSNMEA0
-rw-r--r--  1 root    root     338212 Mar 11 00:59 test.2004-03-10-23.Loadcell0
-rw-r--r--  1 root    root     410400 Mar 11 01:59 test.2004-03-11-00.GPSNMEA0
-rw-r--r--  1 root    root     338118 Mar 11 01:59 test.2004-03-11-00.Loadcell0
-rw-r--r--  1 root    root     409918 Mar 11 02:59 test.2004-03-11-01.GPSNMEA0
-rw-r--r--  1 root    root     338212 Mar 11 02:59 test.2004-03-11-01.Loadcell0
-rw-r--r--  1 root    root     410400 Mar 11 03:59 test.2004-03-11-02.GPSNMEA0
-rw-r--r--  1 root    root     338118 Mar 11 03:59 test.2004-03-11-02.Loadcell0
-rw-r--r--  1 root    root     410400 Mar 11 04:59 test.2004-03-11-03.GPSNMEA0
-rw-r--r--  1 root    root     338118 Mar 11 04:59 test.2004-03-11-03.Loadcell0
-rw-r--r--  1 root    root     410363 Mar 11 05:59 test.2004-03-11-04.GPSNMEA0
-rw-r--r--  1 root    root     338212 Mar 11 05:59 test.2004-03-11-04.Loadcell0


-rw-r--r--  1 gb      users      5753 2004-02-10 15:59 ACUTIME_TEST/test_tb_.2004-02-10-
-rw-r--r--  1 gb      users     19823 2004-02-10 16:59 ACUTIME_TEST/test_tb_.2004-02-10-
-rw-r--r--  1 gb      users     19786 2004-02-10 17:59 ACUTIME_TEST/test_tb_.2004-02-10-
-rw-r--r--  1 gb      users     19819 2004-02-10 18:59 ACUTIME_TEST/test_tb_.2004-02-10-
-rw-r--r--  1 gb      users     19886 2004-02-10 19:59 ACUTIME_TEST/test_tb_.2004-02-10-
-rw-r--r--  1 gb      users     19892 2004-02-10 20:59 ACUTIME_TEST/test_tb_.2004-02-10-
-rw-r--r--  1 gb      users     19867 2004-02-10 21:59 ACUTIME_TEST/test_tb_.2004-02-10-
-rw-r--r--  1 gb      users     20257 2004-02-10 22:59 ACUTIME_TEST/test_tb_.2004-02-10-
-rw-r--r--  1 gb      users     19908 2004-02-10 23:59 ACUTIME_TEST/test_tb_.2004-02-10-
-rw-r--r--  1 gb      users     19935 2004-02-11 00:59 ACUTIME_TEST/test_tb_.2004-02-10-
-rw-r--r--  1 gb      users     19844 2004-02-11 01:59 ACUTIME_TEST/test_tb_.2004-02-11-
-rw-r--r--  1 gb      users     19844 2004-02-11 02:59 ACUTIME_TEST/test_tb_.2004-02-11-
-rw-r--r--  1 gb      users     19855 2004-02-11 03:59 ACUTIME_TEST/test_tb_.2004-02-11-
-rw-r--r--  1 gb      users     19870 2004-02-11 04:59 ACUTIME_TEST/test_tb_.2004-02-11-
-rw-r--r--  1 gb      users     19819 2004-02-11 05:59 ACUTIME_TEST/test_tb_.2004-02-11-
```

```
-rw-r--r--    1 gb        users        19849 2004-02-11 06:59 ACUTIME_TEST/test_tb_.2004-02-11-
-rw-r--r--    1 gb        users        19774 2004-02-11 07:59 ACUTIME_TEST/test_tb_.2004-02-11-
-rw-r--r--    1 gb        users        20147 2004-02-11 08:59 ACUTIME_TEST/test_tb_.2004-02-11-
```

Here below the format of the data files for the GPS sensor is presented. The data were acquired at ISMAR Bologna. The antenna was placed in such a way that few SV were visible. Note the one second update of the ZDA sentence, which is synchronizes to UTC. In contrast, the GGA sentence updates synchrounously with UTC only when the position data were made available (more than 3-4 SVs).

```
$GPGGA,155947,4431.423403,N,01120.369158,E,0,03,2.3,24.36,M,47.51,M,,0000*7C
$GPZDA,160000.73,11,03,2004,-1,00*7C
$GPGGA,155947,4431.423403,N,01120.369158,E,0,03,2.3,24.36,M,47.51,M,,0000*7C
$GPZDA,160001.73,11,03,2004,-1,00*7D
$GPGGA,155947,4431.423403,N,01120.369158,E,0,03,2.3,24.36,M,47.51,M,,0000*7C
$GPZDA,160002.71,11,03,2004,-1,00*7C
$GPGGA,155947,4431.423403,N,01120.369158,E,0,03,2.3,24.36,M,47.51,M,,0000*7C
$GPZDA,160003.71,11,03,2004,-1,00*7D
$GPGGA,155947,4431.423403,N,01120.369158,E,0,03,2.3,24.36,M,47.51,M,,0000*7C
$GPZDA,160004.72,11,03,2004,-1,00*79
$GPGGA,155947,4431.423403,N,01120.369158,E,0,03,2.3,24.36,M,47.51,M,,0000*7C
$GPZDA,160005.72,11,03,2004,-1,00*78
$GPGGA,155947,4431.423403,N,01120.369158,E,0,03,2.3,24.36,M,47.51,M,,0000*7C
$GPZDA,160006.72,11,03,2004,-1,00*7B
$GPGGA,155947,4431.423403,N,01120.369158,E,0,03,2.3,24.36,M,47.51,M,,0000*7C
$GPZDA,160007.72,11,03,2004,-1,00*7A
$GPGGA,160008,4431.400483,N,01120.367407,E,1,04,3.9,28.88,M,47.51,M,,0000*74
$GPZDA,160008.78,11,03,2004,-1,00*7F
$GPGGA,160009,4431.400729,N,01120.367416,E,1,04,3.9,28.81,M,47.51,M,,0000*7F
$GPZDA,160009.75,11,03,2004,-1,00*73
$GPGGA,160010,4431.400968,N,01120.367424,E,1,04,4.0,28.74,M,47.51,M,,0000*79
$GPZDA,160010.73,11,03,2004,-1,00*7D
$GPGGA,160011,4431.401344,N,01120.367441,E,1,04,4.0,28.63,M,47.51,M,,0000*78
$GPZDA,160011.73,11,03,2004,-1,00*7C
$GPGGA,160012,4431.401647,N,01120.367452,E,1,04,4.0,28.55,M,47.51,M,,0000*7A
$GPZDA,160012.71,11,03,2004,-1,00*7D
$GPGGA,160013,4431.401894,N,01120.367460,E,1,04,4.0,28.48,M,47.51,M,,0000*76
$GPZDA,160013.73,11,03,2004,-1,00*7E
$GPGGA,160014,4431.402189,N,01120.367468,E,1,04,4.0,28.39,M,47.51,M,,0000*79
$GPZDA,160014.73,11,03,2004,-1,00*79
$GPGGA,160015,4431.402369,N,01120.367471,E,1,04,4.0,28.33,M,47.51,M,,0000*76
$GPZDA,160015.73,11,03,2004,-1,00*78
$GPGGA,160016,4431.402544,N,01120.367473,E,1,04,4.0,28.27,M,47.51,M,,0000*7B
$GPZDA,160016.73,11,03,2004,-1,00*7B
```

```
$GPGGA,160017,4431.403168,N,01120.367767,E,1,04,4.0,28.48,M,47.51,M,,0000*7E
$GPZDA,160017.75,11,03,2004,-1,00*7C
$GPGGA,160018,4431.403368,N,01120.367774,E,1,04,4.0,28.42,M,47.51,M,,0000*7B
$GPZDA,160018.73,11,03,2004,-1,00*75
$GPGGA,160019,4431.403583,N,01120.367783,E,1,04,4.0,28.37,M,47.51,M,,0000*73
$GPZDA,160019.73,11,03,2004,-1,00*74
$GPGGA,160020,4431.403845,N,01120.367798,E,1,04,4.0,28.30,M,47.51,M,,0000*73
$GPZDA,160020.73,11,03,2004,-1,00*7E
$GPGGA,160021,4431.404077,N,01120.367810,E,1,04,4.0,28.25,M,47.51,M,,0000*77
$GPZDA,160021.72,11,03,2004,-1,00*7E
$GPGGA,160022,4431.404230,N,01120.367816,E,1,04,4.0,28.20,M,47.51,M,,0000*76
$GPZDA,160022.72,11,03,2004,-1,00*7D
$GPGGA,160023,4431.404372,N,01120.367818,E,1,04,4.0,28.15,M,47.51,M,,0000*78
$GPZDA,160023.72,11,03,2004,-1,00*7C
$GPGGA,160024,4431.404491,N,01120.367820,E,1,04,4.0,28.11,M,47.51,M,,0000*7A
$GPZDA,160024.72,11,03,2004,-1,00*7B
```

And here we show the data collected by the Acutime$^{\mathrm{TM}}$. The codes 0x8fad and 0x8fab label the accurate event timestamp and position records, respectively.

```
TSIP_0x8fad,event_count,799,fractional_second,0.6794964328000788,utc_hour,14,utc_minute,42,ut
4,utc_flags,1
TSIP_0x8f0b,event_count,799,time_of_week,225758.6794964328,date,10,month,2,year,2004,utc_offs
,71.90707686077803
TSIP_0x8fad,event_count,800,fractional_second,0.5817191054811701,utc_hour,14,utc_minute,43,ut
4,utc_flags,1
TSIP_0x8f0b,event_count,800,time_of_week,225818.58171910548,date,10,month,2,year,2004,utc_off
t,74.75267158914357
TSIP_0x8fad,event_count,801,fractional_second,0.48394021473359317,utc_hour,14,utc_minute,44,u
44,utc_flags,1
TSIP_0x8f0b,event_count,801,time_of_week,225878.48394021473,date,10,month,2,year,2004,utc_off
t,72.7618990978226
TSIP_0x8fad,event_count,802,fractional_second,0.3861613353656139,utc_hour,14,utc_minute,45,ut
4,utc_flags,1
TSIP_0x8f0b,event_count,802,time_of_week,225938.38616133537,date,10,month,2,year,2004,utc_off
lt,68.6370674725622
TSIP_0x8fad,event_count,803,fractional_second,0.2883760830445681,utc_hour,14,utc_minute,46,ut
4,utc_flags,1
TSIP_0x8f0b,event_count,803,time_of_week,225998.28837608304,date,10,month,2,year,2004,utc_off
,65.94852707907557
TSIP_0x8fad,event_count,804,fractional_second,0.19060268186149187,utc_hour,14,utc_minute,47,u
```

Hereafter follow the perl code to read and plot the GPSNMEA. data.

```perl
#!/bin/perl
# convert ddmm.hh and ddmmss.hh to degree
sub dmhd60 { local ($alfa)=@_;
        local ($segno)=$alfa<0?(-1):1;
        local ($temp)=$segno<0?$alfa*(-1):$alfa;
        local ($xd)=int($temp/100); $yd=$temp-($xd*100);
      return $segno*($xd+$yd/60);
}
sub dmsd60 { local ($alfa)=@_;
        local ($segno)=$alfa<0?(-1):1;
        local ($temp)=$segno<0?$alfa*(-1):$alfa;
        local ($xd)=int($temp/10000);
        local ($yd) = int (($temp - $xd*10000)/100 );
        local ($zd) = $temp - ($xd*10000.0 + $yd*100);
      return $segno*($xd+$yd/60+$zd/3600);
}
while (<>) {
next if not /^$GPGGA/; @A=split(/,/);
$lat=dmhd60($A[2])*($A[5] eq "S" ? -1 : 1) ;
$lon=dmhd60($A[4])*($A[5] eq "W" ? -1 : 1) ;
next if ($lat == $old_lat and $lon == $old_lon);
print "$lon $lat \n";
$old_lat = $lat;
$old_lon = $lon;
}
```

Here follows the Perl code to read the Acutime^{TM} accurate event timestamp data.

```perl
#!/usr/bin/perl
while (<>) {
        if (/0x8fad/) {
                @A=split(',');
$event = $A[2]; $frac = $A[4];
                $hh = $A[6]; $mm = $A[8]; $ss = $A[10];
                $day = $A[12]; $month = $A[14]; $year = $A[16];
                $sfrac = $ss + $frac;
                $sec_since = $day*86400 + $hh*3600 + $mm*60 + $sfrac;
                print "$event $diff $year $month $day $hh $mm $sfrac\n"
                        if $event != 0 and $count > 0;
                $count+=1;
        }
}
```

And finally the bash script that plots the GPS data with GMT:

```
#!/bin/sh
# RTOD, region, projection and Basemap grid,anot and ticks
RTOD=57.29577951308232286465
R=-R11:20:20/11:20:24/44:31:23.5/44:31:27.5
J=-Ju33/1:1000 ; B=-Bg1ca2cf0.1cESwn
PS=GPS.ps
psbasemap $R $J $B -K -P > TMP.PS
# data files TRIMBLE ACUTIME
DATA=`ls ACUTIME_TEST/*.AcutimeEvent `
for d in $DATA ; do
    grep 0x8f0b $d | awk 'FS=","{printf("%.6f %.6f\n", $17*A,$15*A)}' A=$RTOD |\
    psxy  $R $J -Sc0.01 -W0.1/255/0/0 -K -O >> TMP.PS
done
# data files TRIMBLE 4000DL
DATA=`ls *GPSNMEA0.dat`
for d in $DATA ; do
    perl read_GPSNMEA.pl  $d | psxy  $R $J -Sc0.1 -K -O >> TMP.PS
done
echo "showpage" >> TMP.PS ; sed /setpagedevice/d TMP.PS > $PS
```
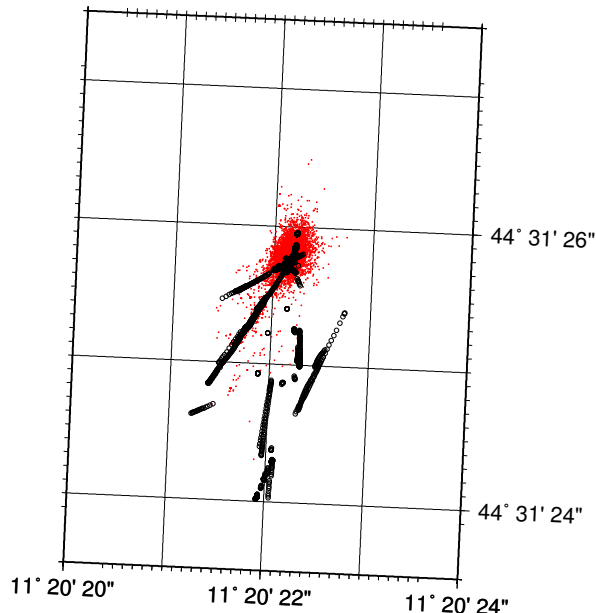
Fig.21 shows the plotted data.

Figure 21: GPSNMEA (Trimble 4000DL) and Acutime[TM] position data plotted.

Herafter an example of Load Cell data follow. The TA5 transmitter dos not provide any internal time information, hence the timestamp is the System clock taken at the time of TA5 query by the software. That timet was also converted to Unix Time.

```
date         ; hh:mm:ss ; unix time       %%%$00+    Load
......
2003-12-07 ; 23:06:13 ; 1070834773646%%%$00+       96 N
2003-12-07 ; 23:06:14 ; 1070834774647%%%$00+       95 N
2003-12-07 ; 23:06:15 ; 1070834775648%%%$00+       96 N
2003-12-07 ; 23:06:16 ; 1070834776649%%%$00+       96 N
2003-12-07 ; 23:06:17 ; 1070834777650%%%$00+       95 N
2003-12-07 ; 23:06:18 ; 1070834778651%%%$00+       95 N
2003-12-07 ; 23:06:19 ; 1070834779652%%%$00+       96 N
2003-12-07 ; 23:06:20 ; 1070834780653%%%$00+       95 N
2003-12-07 ; 23:06:20 ; 1070834780957%%%$00+       96 N
2003-12-07 ; 23:06:21 ; 1070834781958%%%$00+       96 N
2003-12-07 ; 23:06:22 ; 1070834782959%%%$00+       95 N
2003-12-07 ; 23:06:23 ; 1070834783960%%%$00+       95 N
2003-12-07 ; 23:06:24 ; 1070834784961%%%$00+       96 N
2003-12-07 ; 23:06:25 ; 1070834785963%%%$00+       96 N
2003-12-07 ; 23:06:26 ; 1070834786964%%%$00+       96 N
2003-12-07 ; 23:06:27 ; 1070834787965%%%$00+       96 N
2003-12-07 ; 23:06:28 ; 1070834788965%%%$00+       96 N
2003-12-07 ; 23:06:29 ; 1070834789966%%%$00+       96 N
2003-12-07 ; 23:06:30 ; 1070834790967%%%$00+       96 N
2003-12-07 ; 23:06:31 ; 1070834791968%%%$00+       96 N
2003-12-07 ; 23:06:32 ; 1070834792969%%%$00+       96 N

....
```