Environmental Control System based on mobile devices

T. Albero maalal0@upvnet.upv.es
V. Sempere vsempere@dcom.upv.es
J. Silvestre jsilves@disca.upv.es
P. Dabbas patdabca@teleco.upv.es
Technical University of Valencia (UPV)
Camino de Vera s/n
Valencia. Spain

Abstract-

The impact of the use of mobile devices by means of GPRS (General Packet Radio Service) technology to access environmental information in real time, provided by sensors strategically placed in a metropolitan area has been studied with the objective of using this information to find out rainfall volume and phreatic levels in times of emergency. This allows the technical services to make the correct decisions in the management of a large network of pump stations, flood gates, etc. It is demonstrated that this is a reliable system with an acceptable response time. The system being tested is integrated into a powerful supervision and control system that works in the purification network of the city of Valencia. It has been developed jointly by the "Ciclo Integral del Agua" and the Technical University of Valencia.

***Index Terms*—Sensor networks, Control, Supervision, Monitoring, Environmental.**

1. Introduction

In the current control and management systems of water purification networks, it is clearly necessary to obtain and process different information [1, 2] in real time from sensors located over a large geographical area. These sensors provide information on rainfall, water quality [3] (coasts, rivers, lakes, marshes, etc.), phreatic levels, etc. Fixed networks don't always cover all the areas where the measurements have to be carried out, and so, systems that allow information to be obtained quickly and at low cost must be designed. In this study the results using GPRS (General Packet Radio Service) devices for the transmission of information captured by different sensors to a Central Control System in real time is shown.

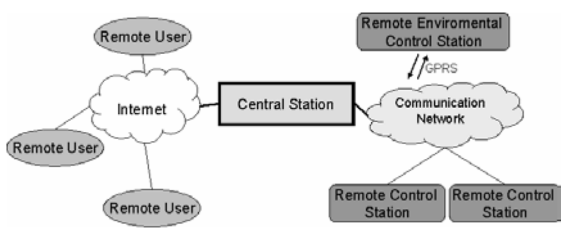

Fig. 1 General scheme of the Supervision and Control System

In section 2, a global system is described briefly, which houses two types of remote station, one of which, the environmental station, is the subject of this paper, and is described in section 3. In section 4 the experiment and the results obtained using GPRS as a way of transmitting the information obtained by the sensors are shown. Finally, in section 5 the conclusions of the study are shown.

2. Architecture of the System

The "Controlvision" System is a Supervision and Control System for purification networks that consists of four fundamental parts: a Central Station, Remote Stations, remote users that connect to the system via Web and a Communication Network, see Fig. 1.

The Central Station manages the information from the installation and also provides services for remote access. The Remote Stations allow information from different devices to be obtained, whether they are cameras, automatons or instrumentation devices.

The remote users connect via Web to the system with different devices: PC´s, laptops, PDA´s (Personal Digital Assistant), mobile telephones, etc. The communication Network gives the support for the communication between the Central Station and the Remote Stations.

2.1. Central Station

This is composed of various elements: a Central Control Station with which it is possible to access the whole system, a data base where the information and the configuration is stored, a Web server for remote access via Internet, a Firewall that provides security against external attacks and finally the Data Network, which facilitates communication between the different elements of the Central Station. The interconnection of these elements is shown in Fig. 2.

The Central Control Station enables the operators that are physically in the Central Station to view camera images (in real time or stored), consult data from the PLC´s (Programmable Logic Controller) of the Remote Control Stations and the data from the Remote Environmental Control Stations. Windows 2000 Server has been used as the operating system and the

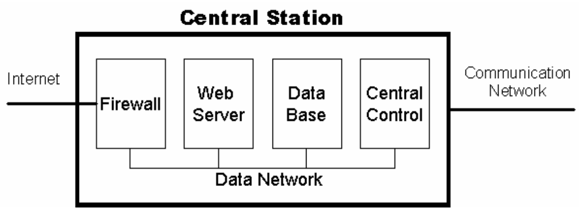

Fig. 2 Plan of the Central Station

programming languages Borland C++ Builder and Microsoft Visual C++ for the user interface application.

Another machine with Linux houses the data base (MySQL) which stores information from all the system that the Central Station provides (videos path, devices states, instrumentation data, remote stations configuration, users and kind of users, etc).

The Web Server provides images, information on the status and data on the instrumentation devices to the remote access users, making it easy to access the information from the system whatever the user's location. Access by means of the Web is restricted, requiring authentication, and there are different authentication levels. Depending on their privileges users can access specific contents. The operating system used in this equipment is GNU/Linux which provides the Apache Web Server (Second edition) with support to PHP4 (Personal Home Page, Fourth edition). The Web is programmed in XHTML (Extensible HyperText Markup Language) following the Standard of W3C [4] (World Wide Web Consortium), this way it will be accessible from any operating system and from any explorer.

For the Network data a GEthernet switch with support to 802.1p has been used. The implementation of the Firewall is realized with GNU/Linux by means of IPTables. This is a public license solution, highly configurable and robust. It will only permit the information coming from the essential ports to enter, stores any event that could be considered unusual, and can send an e-mail to the Administrator advising him of any anomaly.

2.2. Remote Stations

These can be of two types: the remote control stations and the environmental control stations. The remote control stations include: the pumping stations, the underpasses (in these two types of remote stations an automaton and/or cameras exist), the lock gates and the purifying stations. In these stations PC´s with Windows XP Professional which are used as a gateway are installed.

The Environmental Control Stations are stations with instruments that obtain data from different sensors such as: flow meter (measuring instrument specializing in water measurements), rain gauge, dissolved oxygen probe (measures the amount of oxygen in the water), conductivity probe (device to measure the PH-temperature), ultrasonic sensor (measures the level of water), piezometer (equipment that measures the phreatic

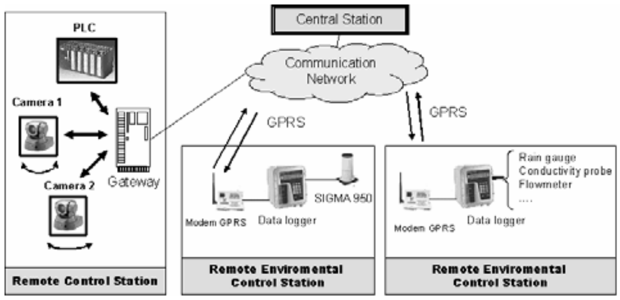

Fig. 3 Different kinds of Remote Station with their equipment.

level, that is, the height of the uppermost aquifer subterranean layer), etc.

All the information from the sensors is obtained by a data logger and can be read through the port RS-232 by a modem GSM/GPRS which executes our program in J2ME (Java 2 Platform, Micro Edition) with the protocol used by the data logger, and which transmits by means of TCP/IP via GPRS to the Central Station to communicate with the J2ME program in the modem. An example of two types of Remote Station is shown in Fig 3.

2.3. Communication Network

The Communication Network provides communication to all the Remote Stations and allows the incorporation of new real time services. Different technologies have been evaluated and presented in other previous Works. In [5] different communication alternatives were studied over public Networks. In [6] remote access by means of a Web client by GPRS to obtain images and control information from the Remote Control Stations were evaluated, concluding that it is a good way to obtain control information at any moment and from any location. In [7] a comparison between UMTS (Universal Mobile Telecommunication System) and GPRS for access to the information by means of mobile devices is shown. In this study the conclusion obtained was that in Supervision and Control Systems GPRS with a mobile device such as a PDA is useful for consulting at regular times whereas UMTS with a laptop permits the monitoring of images and data in real time

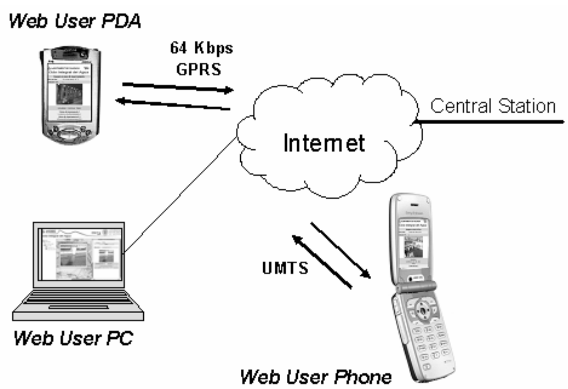

Fig. 4 Various remote users that can connect to the system.

with a good response time.

2.4. Remote users

These are the authenticated users that connect to the System via Web obtaining information according to the authentication level they have. This information can be: images in real time, stored images, equipment states, instrumentation information, etc. A wide variety of devices is used to make the connection, as shown in Fig. 4.

3. Environmental Control Stations

3.1. General description

The Remote Station will be composed of a modem and a data logger with the sensors enumerated in section 2.2. Next, some characteristics of these two devices are described.

The Siemens' modem TC45 [8] GSM/GPRS is used, with a CLDC (Connected Limited Device Configuration), and a reduced version of MIDP1.0 (Mobile Information Device Profile), called IMP 1.0. MIDP is provided by Sun, and IMP is provided by Siemens's. IMP 1.0 does include a special API (application program interface) for handling files, a specific API for AT commands (the de facto standard language for controlling modems), and another for serial communications, and a specific API for TCP/IP connections such as Datagram Connection, HTTP (HyperText Transfer Protocol) Connection and Stream Connection. The latter is used in this project for socket TCP implementation.

The modem is programmable by J2ME [9], a programming language developed by Sun, specially designed for small embedded, consumer devices, including mobile phones, PDA (Personal Digital Assistants), and a wide selection of mobile and cordless devices. Sun One Mobile Edition [10] was used for the programming and debug of the J2ME program. The modem has two interfaces: one connected to the data logger (serial connection), and the other, which will interact with the Central Station (GPRS). GPRS has been implemented over the TCP/IP sockets, which are supported by the modem.

The data logger is a device that reads and writes in its memory the data from the different sensors connected to it. The readings can be made of real time values or stored values. The memory used by the data logger allows 115,630 data logs in total, which is for all the different sensors. The data logger's behaviour when the memory is full is programmable. We can decide if the data logger stops taking measurements, or make the data logger work cyclically, erasing the oldest measurements. The periodicity of the measurement can also be chosen from among the following: 1, 2, 3, 5, 15, 30 or 60 minutes. The data logger has a serial port to be accessed for the readings, and permits an access up to 19,200 bauds.

The basic lines of work for this system are: the data logger reads the sensors periodically and saves the values in its memory, so that it can be read afterwards. The modem will have the GPRS connection opened connected to the Central Station waiting for the latter to make a request. The modem will then translate these requests to the data logger through its serial port. The data logger's answers will be directly sent to the Central Station without any type of processing, to free the modem of computational load.

For this type of station, which normally are in areas where there is neither coverage nor electricity current, it is necessary to take into account the life of the battery where the devices are connected. This time is important because it will be necessary go to the station to recharge the battery. Currently, batteries that are recharged every 2 months are used to supply the system.

To obtain the data in the Central Station from the data logger two applications have been designed; one that will run in the Central Station, and the other one running in the mobile device. This was necessary because most of the data loggers do not have a program to download the data that can be installed in a mobile device (the modem in this system), and if any, they would need a specific program running in the Central Station, therefore our System would not be 100% compatible. We need a simple program on the Central Station capable of controlling all the Remote Stations, not only the Environmental ones, and to adequate the system to the requirements of the technical people who are using it, and consequently a standard program cannot be used.

There are other solutions such as the applications that exist on the market that can be used for several data loggers, but these are only valid if a standard protocol such as MODBUS [12] is used. Moreover to be able to download data from the logger, the application must be installed in a PC. This solution was ruled out because the logger in use does not support MODBUS, and it is not possible to install a PC in the Remote Station.

Another solution is to use the application of the manufacturer, but this solution has the drawback that the application must be installed in a PC and it needs to be controlled remotely; which is complicated from the point of view of the worker who needs to see the values in real time from different stations at the same time.

Some companies offer a modem that converts RS232 to Ethernet, for example Temple with its module Moxa [13] and then transmit the data via Ethernet. This option was ruled out because it can not take wired Ethernet, or any other wired connection to the location of the environmental stations. Another option offered by the same company, is to use the converter, and a GPRS router. This solution has a high price.

There are applications on the market as well that permit transmission by GSM, like the modem GSM WM02 [14]. Using the traditional technology of POOL GSM, GSM calls have to be made to the remote devices when some data has to be obtained. Normally this is done by making a sequence of calls. The cost of devices is relatively low, since only modems are needed, but the configuration and preparation of the modems is much higher than with the GPRS modem. Furthermore, if we want readings for different stations in parallel, a pool or a

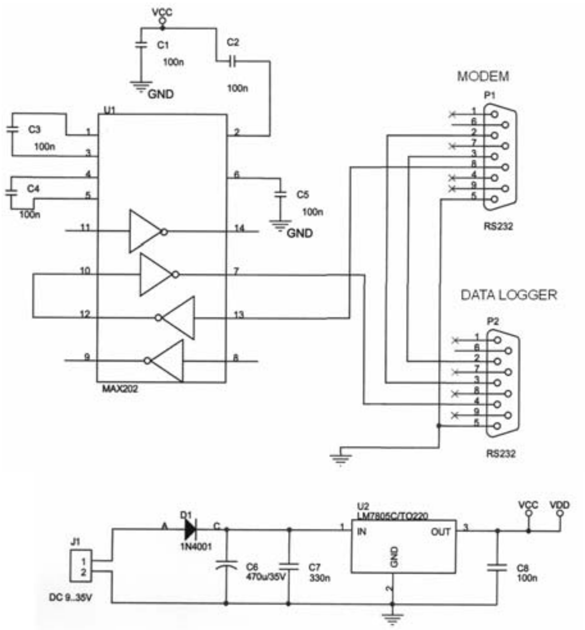

Fig.5 Scheme of the serial connection between the modem and the data logger using two male DB9 connectors

line for every station is needed. With the design shown in the paper, all the data can be obtained simultaneously and this is very favourable. Therefore, the option of GSM was ruled out because of the price and the small scope.

None of these solutions are valid in the case of the Environmental Control Stations studied due to problems space, hard accessibility, adverse meteorological conditions and low availability of electricity network because they are on the outskirts. Therefore, the only solution is a small device, using battery and a communications system that doesn't need prior installation, and moreover that allows us to programme the serial communications protocol. Because of this, we have developed our own application. The application must on the one hand communicate with the data logger using the adequate protocol by the serial port, and on the other hand must communicate with the Central Station via GPRS. Therefore we obtain a completely compatible system that is easily integrated with the rest, and that is easily reprogrammed to use almost any logger in the market. This way, the network in the city does not have to be changed, and the loggers already installed are used.

3.2. Communication Modem – Data logger

The modem has two serial ports called ASC0 and ASC1. The ASC0 is a RS232 complete interface that is used to control the modem using AT commands, or as the standard data output stream, used for debugging [11]. The second serial interface, ASC1, is a 4-wire interface (RX, TX, RTS, CTS and GND). This interface is the one used to connect the modem to the data logger.

The data logger's serial interface is a 4-wire interface (RX, TX, DSR, DTR and GND), which means that both interfaces cannot be connected directly.

The MODEM supports the CTS/RTS protocol. When it is ready to transmit, it put the CTS line to logic "1". At first, we tried to connect this line to the logger's DSR, but the logic "1" was not maintained, and therefore both devices could not communicate. The solution adopted was to design and insert a circuit which is capable of maintaining the logic "1". We used the integrated circuit MAX202.

In Fig. 5 we can see the connections necessary for the correct communication between the data logger and the modem. We can observe that the modem's CTS is connected to pin 13 of the MAX202, and the logger's DSR is connected to pin 7. The pin numbers 12 and 10 have been short circuited. This way, the circuit MAX202 maintains the necessary level for the serial communication. The MAX202 converts from pin 13-12 a RS232 to a TTL (Transistor-Transistor Logic), and then the 10-7 connection makes the opposite conversion, TTL-RS232.This way we assure a valid level for the serial communication. The 5V supply needed by the integrated chip is obtained from the 12V supply the data logger has, using a "regulator".

When the data logger answers, it is not necessary to put the RTS to "1", because we make the modem read at a specified time by software.

The data logger uses a simple Stop&Wait [15] protocol, where most of the communication uses ASCII characters. The different requests that the data logger admits are:

I. Identification Request. The data logger will respond with its own identification number, data logger model, date and time of the request.
II. Configuration Request. The Data logger will answer with its identification number, version of the software installed, units of measurement and time gap used between measurements.
III. Event Request. The Data logger keeps a record of significant events which occur, such as start, end or pause in recording measurements. This request will enable us to download all events which occur.
IV. Current Request. The response for this request will again be its identification number, date, hour and this time we will also be able to read the last measurement the data logger has for all the sensors attached to it, as well as the battery level.
V. Device Request. The response to this request is information about all the sensors that it controls and the characteristics of these, such as name, identification number, number of measurements recorded, date and time of the beginning of the recording, and time gap between measurements.
VI. Read Recorded Data. This is a simple request in which we must specify the sensor we want the data from, and the data logger will write all the data recorded for the sensor specified.
VII. Read Recorded Data since last Download. With this request we will only download the new data recorded since the last access to the data logger. It is the data logger the one who records the last time data was read from each sensor.

Except in the case of the transmissions of numeric values in the data tables (Instructions VI and VII), the

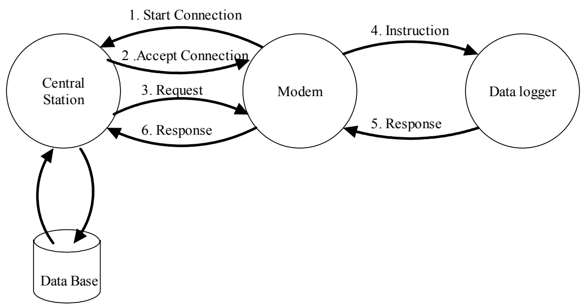

Fig 6 Data Flow Diagrams for the general system and Environmental Control Station.

codification is immediate. Each byte in hexadecimal corresponds to an ASCII character, in the instruction request as well as in the answers given by the data logger. The data tables start with 3 bytes, followed up by the measurement coded using Standard IEEE 754 simple precision (32 bits), and rearranged from least to greatest significant values. And finally the data table has a checksum using two bytes.

An important aspect to take into consideration is the response time of the data logger. The data logger used can take up to 12 seconds before answering an instruction. The time elapsed until the data is ready to be read on the serial interface, varies with the information (bytes) actually written. It has been necessary to introduce a waiting time from when the modem writes the request until it tries to read the answer. This has had to be done by software. A different time in seconds has been given for each type of request.

3.3. Modem-Central Station communication

The modem is the client on our application, and is connected with the Central station with an IP variable by means of GPRS. GPRS is implemented over TCP sockets, obtaining in this way better reliability than with UDP. Once the connection is accepted, a TCP socket will be open connecting the Central Station and the modem, to which all requests and responses will be written, and read from. Whenever the Central Station makes a request to the modem, the latter will start the communication protocol with the data logger. Immediately after the modem receives the response, it will write it on the TCP socket without any type of data processing. The modem acts as a simple interface, doing no more work than necessary. This is because our modem is a limited device, and we do not want to overload it with unnecessary work. A Data flow diagram is shown in Fig. 6.

The modem will automatically connect to the Central Station, and remain connected. If the modem detects that the socket between itself and the Central Station broken, it will retry to reconnect. After a number of failed reconnections, it will assume it is not connected to GPRS, and attach itself again. After this reattachment to the GPRS, it will try again to reconnect to the Central Station. To keep the socket TCP open, it is necessary to send one byte every 30 seconds. This is not a lot of bytes throughout the month, sending 1 byte each 20 seconds, which translates to 4320 Bytes each day and 129600 Bytes (130 Kbytes) per month. With a price of 0.001 €/Kbytes per month the total costs are 0.13 € per month (with the Spanish operator Amena) to avoid the socket between the modem and the Central Station being closed.

In Fig. 7, we can follow the steps taken from when the Central Station makes a request for data, until it receives

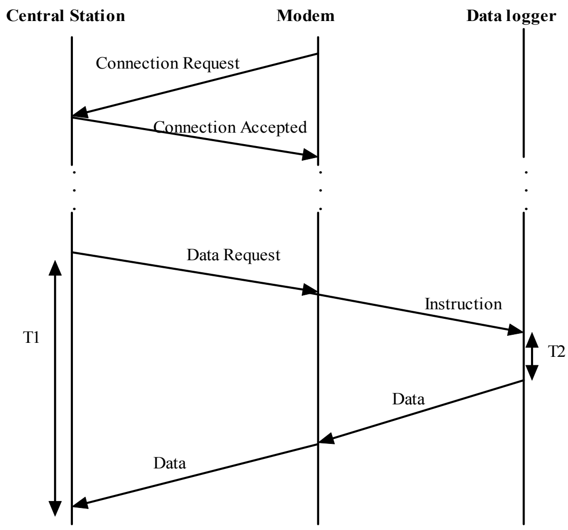

Fig 7 Sequence diagram of a request from the Central Station to the MODEM to obtain information from the measurement device, corresponding to instruction numbers I, II, III, IV and V.

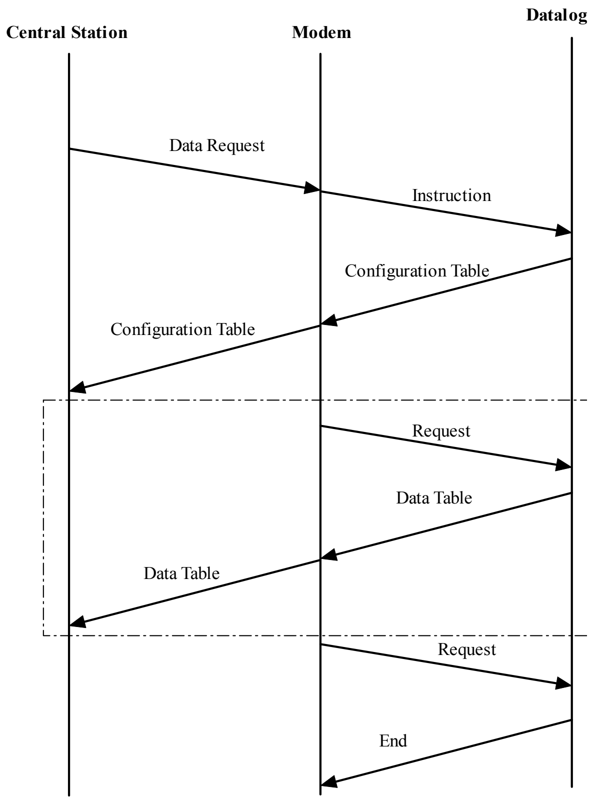

Fig. 8 Sequence diagram of a request from the Central Station to the MODEM to obtain data from the data logger, corresponding to instructions VI and VII.

the information it wants. These steps correspond to instructions I, II, III, IV and V explained previously in this document. Once the modem is connected to the Central Station, the Central Station can request all types of information from the modem, which will transfer these requests to the data logger. The data logger will respond with information tables which are then sent to the Central Station for further processing and decoding. The data logger-modem communication is very simple, only a little data coded in ASCII is transferred between them.

In Fig. 8 a more complete protocol is shown: the one used on requests VI and VII. Unlike the last scheme, we observe a more complex communication between the modem and the data logger, which was not necessary before. Once a connection is established between the modem and the Central Station, the latter makes a request to the modem. This will write the instruction on the data logger serial interface. The first response of the data logger will be an information table, coded in ASCII, about the sensor from which we are about to read the measurements. This information table is immediately sent to the Central Station.

This is the table named Configuration Table in Fig. 8 Some of the information we get from this table is: sensor ID, sensor's name, date and time of start of the measurement, number of measurements recorded on memory, and time gap between measurements. Once the modem has sent this configuration table, it is ready to read the measurements. The modem informs the data logger of this condition, waits the specified time, and reads the serial port. This time the data logger answers with data tables with measurements. These tables are no longer coded in ASCII, but using IEEE 754 simple precision coding as specified previously in this article. In each table, the data logger can write up to 256 pieces of information. Therefore the number of tables received depends on the number of measurements we read. Each time the modem receives one of these tables, it is transmitted immediately to the Central Station, and continues with the protocol to keep on reading more data tables, until the data logger sends the end character because all the data stored has been read, and therefore no more measurements are left. Reading of the measurements does not erase these from the memory. Once the modem receives the end of transmission character, it closes the serial communications and waits for more requests from the Central Station.

The modem is ready for initialization of the global reading program even when the electricity supply is cut off. In such case the J2ME program that starts the GPRS program will initiate automatically, and therefore the modem will reconnect to the server. This permits better efficiency, and avoids the transfer of a specialized member of the crew to initiate the modem.

4. Tests and Results

Time measurements have been made using instruction IV (Current Status), which gives us the last measurement the data logger has for all the sensors attached to it, along with basic information from the data logger. The time measured is the time elapsed from when the Central Station makes an inquiry to the modem asking for its current status (instruction IV), until the Central Station gets this information. The request is required every 15 minutes using a SIM card of the Spanish company Telefonica Moviles, lasting 10 days.

Table 1 Data given by the data logger when asked for a real time measurement reading.

```
CURR:
0010:13 25-FEB-05
01F3Y
02IDENTIDAD LUGAR: 00000007
03
04EN EJECUCION
05NIVEL: 70.361cm
05FLUJO: 0.935cms
05TOTAL(x1000): 935.m3
05VELOCIDAD: 0.74m/s
 05ALIMENTACION:.15.9.VOLTS
DONE
```

This instruction will always respond with the same amount of bytes, depending on how many sensors the data logger has attached to it. In our test, we used a data logger with two sensors: velocity and level sensors. With these two measurements, the data logger is able to calculate the water flow. The data logger used for this testing responds with Table 1 coded in ASCII:

In this table we have 205 bytes. The table shows us that the measurement was taken 25.2.2005 at 10:13, that the data logger has the number 00000007 as an identification number. The next line, starting with 04, indicates the state of the data logger, in this case EN EJECUCION means that the data logger is at the moment reading measurements and recording them in the memory. The next row starts with 05, these are the last measurements taken. In this case we have a measurement for level, velocity, flow and battery. As mentioned above, it is instruction IV, and therefore follows a transmission model as in Fig. 7. The data logger is programmed to take measurements every 5 minutes. There is no lost information, even though we are reading every 15 minutes. All of the measurements are written in the memory, read or not by this instruction or any other. We are reading the current status of the data logger every 15 minutes, but at any moment we are able to read the measurements history, by simply making another type of instruction, such as VI or VII.

As shown in Fig. 7, T1 is the time taken from when the Central Station makes a request until the response reaches the Central Station. This time can be divided into two major contributions; first, the time the response takes to travel over GPRS (TGPRS) and second, the time the data logger takes to answer (T2). T2 is specified by software in the modem program, and depends only on the data logger's response time. For instruction IV, this time is 10 seconds. This is a constant value that will not change from one reading to another. Assuming that the execution time on the modem is zero, the GPRS time (TGPRS) will be T1-T2=TGPRS.

Fig. 9 shows the variations of T1, during the 10 days the test was done. We can appreciate many fluctuations. We have already said that T2 is a constant value; therefore the same peaks are shown in Fig. 10 which represents the GPRS time only. Therefore the peaks shown in Fig. 9 and Fig. 10 are due to GPRS. We can appreciate it is not constant; there are many peaks to take into consideration, though the average is 2.01 seconds which is a good time. However, this value is enclosed in a reasonable maximum. Furthermore, taking into account that the data loggers are making readings from the sensors every 5 minutes, a delay of 15 seconds maximum, as shown in figure 10, can be considered negligible We are not dealing with a hard deadline, therefore this delay, although not welcome, does not reduce the effectiveness of the system.

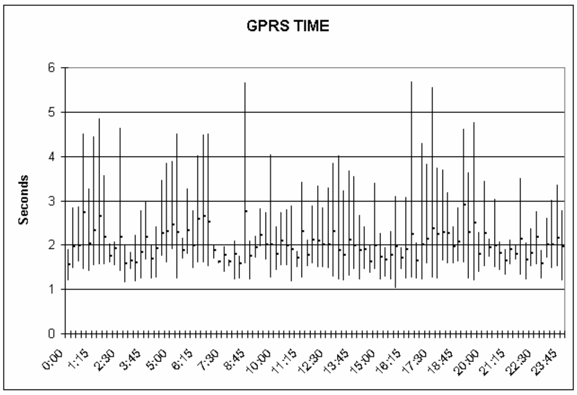

Fig 9 GPRS time

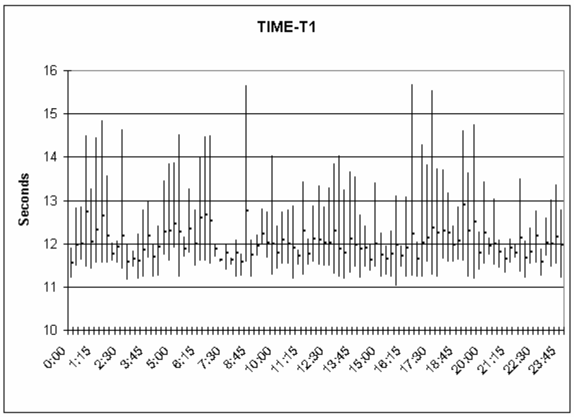

Fig 10 Response time for the interconnection using GPRS

5. Conclusions

In the study, the total time that is taken to receive data from when the central station asks for it, depends on the time the data logger delays in answering and the GPRS time that has been shown.

As has been observed, the GPRS response time is more than acceptable, allowing the whole system to achieve the efficiency required. The other time, the response time of the data logger is fixed and specified by software, and therefore is a controlled time. Having both times enclosed and known, we have a predictable and therefore feasible system that allows quick decision making.

Moreover, the choice of GPRS is justified when the average bytes transmitted in a month is studied. By our reckoning, we have estimated 1Mbyte/month will be transmitted by a data logger. This value has already taken into account the environmental data and the bytes that have to be transmitted every 30 seconds to maintain the modem-Central Station socket opened. Therefore the amount charged by the Spanish operator Amena, would be approximately 6 €/month, per Remote Station. This cost is not too high and justifies the choice GPRS as the technology for the transmission at it is a cheap system, and has no need of previous installation of any kind in the Remote Station, which is already established as not possible.

With the tests we have observed how, with a simple J2ME program, we can have a real time remote measurement system, which is flexible and reliable.

The processing of data is done in the Central Station; this opens a wide range of opportunities over the global system. We also free the modem, which acts as a simple interface between the Central Station and the data logger, of computational load. The modem, therefore implements a simple Stop&Wait protocol to communicate with the logger, easy to program, and which does not need much resources. This will allow use of the same system to similar devices controlled through their serial port, and for different loggers from different manufacturers, with different protocols for transmission of the data, because this can be easily programmed using J2ME in the modem.

In conclusion, the system analysed in this study, is useful, and recommended for remote reading of measurement, and real time readings of similar devices.

6. Acknowledgements

This work was supported by the "Ministerio de Ciencia y Tecnología" of Spain under the project TIC2003-08129-C01, which is partially funded by FEDER and by the “Generalitat Valenciana” with the grant TS/03/UPV/09.

7. References

[1] Rob Holman, John Stanley and Tuba Özkan, “Applying Video Sensor Networks to Nearshore Environment Monitoring”, IEEE Pervasive Computing Mobile and Ubiquitous Systems. Vol 2, Nº 4, pp. 14-21. October-December 2003.

[2] A. Del Grosso, F. Lanata, D. Inaudi, D. Posenato, A. Pieracci, "Breakwater deformation monitoring by automatic and remote GPS system" in The First International Conference on Structural Health Monitoring and Intelligent Infraestructure, November 2003, Tokyo, Japan.

[3] Jerry C. Ritchie and Charles M. Cooper, "Remote Sensing Techniques for determining water quality: Applications to TMDLs" in TMDL Science Issues Conference, Water Environment Federation, Alexandria, VA 2001. pp. 367-374.

[4] http://www.w3.org/

[5] V. Sempere, T. Albero, J. Silvestre., "Analysis of Communication alternatives over Public Networks for a Supervision and Control System of Metropolitan Scope", in Proceedings of the IEEE International Symposium on Industrial Electronics, 2004, ISBN 0-7803-8304-4 pages 597-602.

[6] V. Sempere, T. Albero, J. Silvestre. "Remote Access to Images and Control Information of a Supervision System through GPRS", in Preprints of the 1st IFAC Symposium on Telematics Applications in Automation and Robotics, 2004, 107-102.

[7] V. Sempere, J. Silvestre, T. Albero, C. Díaz, "Acceso Remoto a Imágenes e Información de control de un Sistema de Supervisión mediante dispositivos móviles", "Automática e Instrumentación", no. 357, December, 2004, 64-69.

[8] TC45 JAVA User's Guide. Siemens Cellular Engine. Version: 03. DocID: TC45 JAVA User's Guide_V03.

[9] Kim Topley, *J2ME in a Nutshell: a desktop quick reference,* O'Reilly, Sebastopol: 2002, 450 pages. ISBN 059600253X.

[10] Sun™ ONE Studio 4, Mobile Edition Tutorial. Sun Microsystems, Inc.

[11] TC45 Siemens Cellular Engine. Version: 02.02. DocId: TC45_ATC_V02.02.

[12] www.modbus.org

[13] http://www.tempel.es/www_gprs_tempel_es/

[14] http://www.omniinstruments.ie/catalog.0.html4.1.html

[15] Andrew S.Tanenbaum "Redes de Computadoras", Tercera Edición. Prentice Hall.