Measurement 34 (2003) 75–83

Measurement

www.elsevier.com/locate/measurement

Development of a data acquisition system for remote monitoring of renewable energy systems

Kostas Kalaitzakis*, Eftichios Koutroulis, Vassilios Vlachos

Department of Electronics and Computer Engineering, Technical University of Crete, GR-73100, Chania, Greece

Received 27 January 2002; received in revised form 14 February 2003; accepted 20 February 2003

Abstract

The widespread application of Renewable Energy Sources (RES) requires the use of data acquisition units both for monitoring system operation and control of its operation. In this paper, the development of a data acquisition system for remote monitoring and control of RES plants is presented. It is based on the Client/Server architecture and it does not require the physical connection of the monitored systems to the data collection server. This feature is essential in RES plants since they are usually installed in inaccessible areas. The measured parameters are available on-line over the Internet to any user.

Keywords: Distributed measurement system; Renewable energy sources; Monitoring; Internet; Java; Data acquisition

1. Introduction

The recent growth of networks and especially the spread of the Internet have boosted the development of distributed measurement systems for a variety of applications. Most distributed measurement systems follow the Client/Server architecture [1,2], as outlined in Fig. 1. According to this architecture, one or more instruments are connected to a measuring station, which operates as a server, while the acquired data are available through a network to the clients.

Regarding power networks monitored by SCADA systems, distributed measurement systems are used for remote access to data and control [3], as illustrated in Fig. 2a. The data distribution system is based on the Client/Server architecture, while the data transmission is carried out through the Internet. A similar application is proposed in Ref. [4], where a master monitoring station is used to collect power and current related data from slave stations. Each slave station monitors the operation of a power system and transmits the collected data to a server station, through a serial port interface, as depicted in Fig. 2b. The clients retrieve the data through an Intranet network.

Another interesting application area of distributed measurement systems is the remote control of instruments [5], as illustrated in Fig. 3. According to this topology, a number of instruments are connected to a server station using standard interfaces such as the VXI-11, the IEEE-488 and the RS-232 protocols. The clients communicate with the server using the

*Corresponding author. Tel.: +30-28210-37213; fax: +30-28210-37530.

E-mail address: koskal@electronics.tuc.gr (K. Kalaitzakis).

0263-2241/03/$ – see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S0263-2241(03)00025-3

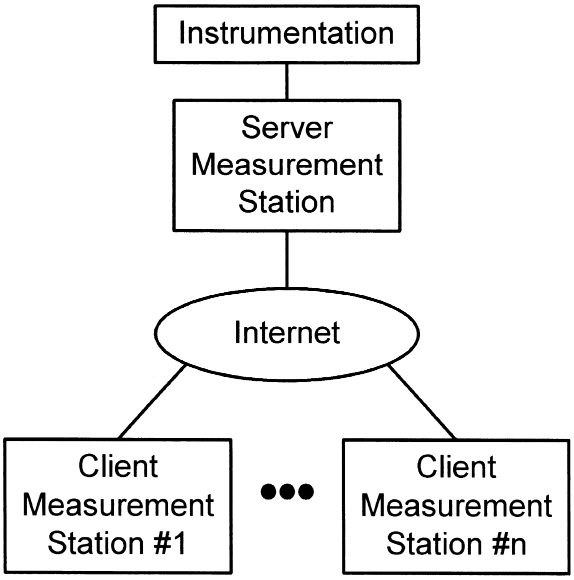

Instrumentation
Server Measurement Station
Internet
Client Measurement Station #1
●●●
Client Measurement Station #n

Fig. 1. Distributed measurement system based on Client/Server architecture.

Internet. In an alternative configuration shown in Fig. 4, a number of sensors and actuators are connected to network nodes using the IEEE-1451.2 standard interface, while the network nodes communicate with each other by means of an Ethernet network and a gateway interconnects the Ethernet with the Internet by means of the TCP/IP protocol [6]. The client applications monitor and control the instrumentation through the Internet. A different approach is proposed in Ref. [7], where the data collected from a sender are transmitted to receivers over a network using a real-time protocol [Real-time Transport Protocol/RTP Control Protocol (RTP/RTCP)] in order to provide guaranteed quality of service (QoS).

A different distributed measurement systems concept is networking of remote laboratories [8], mainly developed for teaching purposes, where several measurement laboratories, distributed over a wide geographic area, are linked through the Internet, as shown in Fig. 5. The students have access to the instruments over the Internet through the General Server (GS), which in turn communicates with the corresponding Local Controller (LC_i). The last forwards the request to the Measurement Controller (MC_j) responsible for performing the requested measurement.

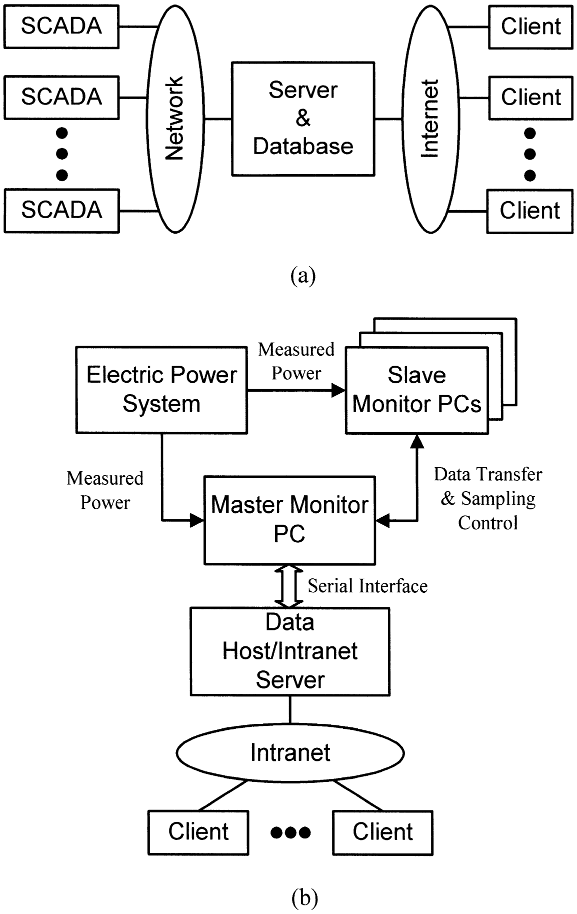


Fig. 2. Distributed measurement systems for power networks: (a) Internet-based data transmission and (b) data transmission using an Intranet.

A common characteristic of all distributed systems described above, is that the process or instrument under control must be physically connected to the Server Measuring Station via either a network, a serial communication port or a standard interface (e.g. IEEE-488).

Nowadays, the wide application of Renewable Energy Sources (RES) requires the use of data acquisition units for both monitoring the system operation and evaluating performance [9]. An important characteristic of RES systems is that they are

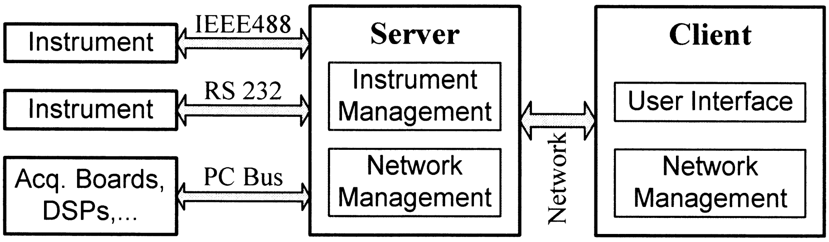


Fig. 3. Distributed measurement system for remote control of instruments.

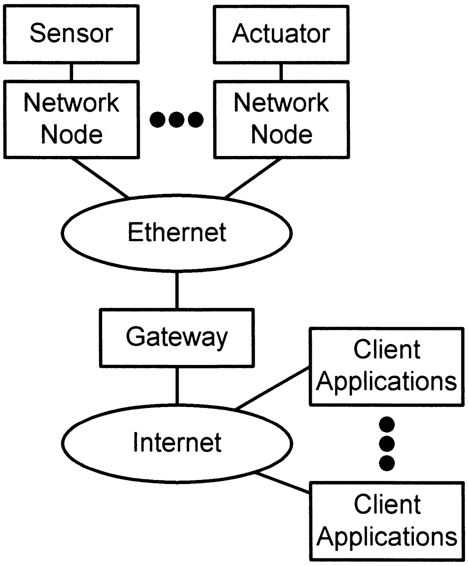


Fig. 4. Distributed measurement system with sensors and actuators connected to network nodes using the IEEE-1451.2 interface standard.

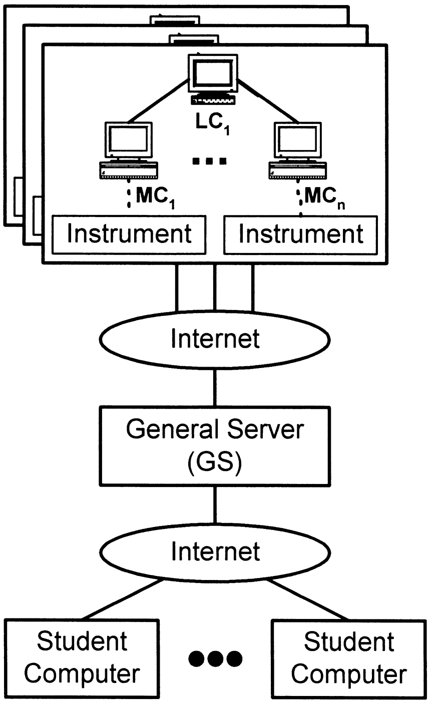


Fig. 5. Distributed measurement system for teaching purposes.

usually installed in geographically remote areas, where telephone lines and computer networks are, in most situations, not available. Furthermore, in the case of large RES plants, control of output power is of major importance, due to power system stability and safety requirements. The RES data acquisition units developed so far [10,11] are designed only to store acquired data either on a local computer or on a remote database, usually via a telephone line, where data from several other RES systems are also stored.

In this paper, the development of a data acquisition system for RES systems remote monitoring and control is presented. The proposed system architecture is illustrated in Fig. 6. All interesting signals at every remote RES station, such as photovoltaic voltage and current, irradiation, wind speed, etc. are measured with a local data acquisition unit and the acquired data are transmitted to the corresponding RES-Collector unit by means of an RF link and a serial port. Sequentially, the data are transmitted to the main Server unit (RES-Server) using an Ethernet

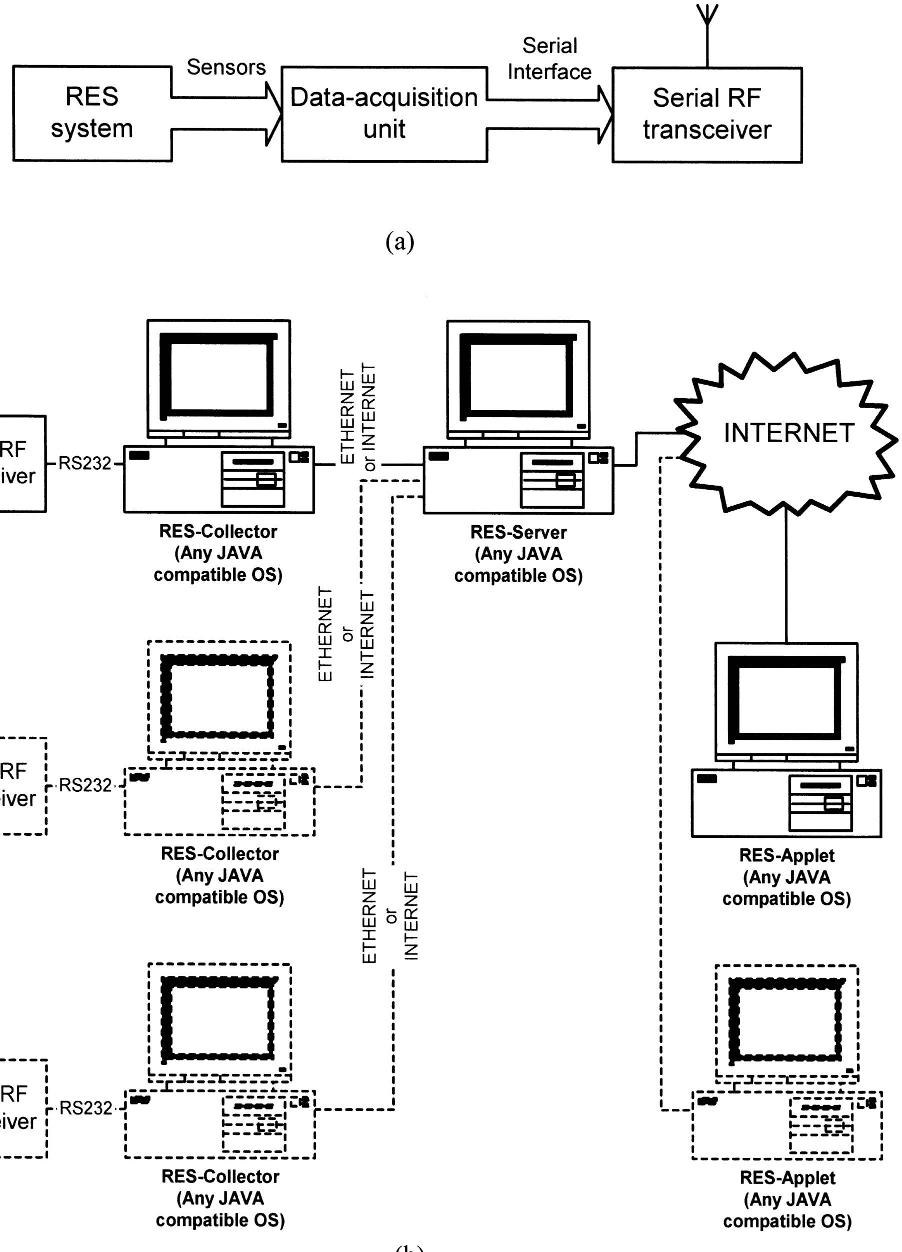


Fig. 6. The proposed system diagram: (a) the data acquisition unit and (b) the distributed data networking system.

type network or the Internet. The clients can retrieve the data from all RES stations through the Internet, using the developed Applet interface. The proposed system architecture is described in detail in Section 2, while a laboratory implementation of the proposed system is described in Section 3.

2. The system architecture

The proposed system block diagram is depicted in Fig. 6 and consists of four parts:

- A number of data acquisition units (Fig. 6a), each connected to a set of sensors mounted on the RES system collects the desired data and communicates with the corresponding RES-Collector via an RF transceiver.
- A number of computers each equipped with: (a) an RF transceiver connected to its RS-232 port and (b) an interconnection to an Ethernet network or to the Internet. A program responsible for the collection of data from each RES station (RES-Collector) runs on each computer (Fig. 6b).
- A computer interconnected to the Internet, running a program which collects data from all RES-Collectors and properly routes them to the users (RES-Server) (Fig. 6b) and
- A number of users' computers interconnected to the Internet, which display the data using any browser, by means of the presentation interface (RES-Applet) (Fig. 6b).

For universality reasons, all software running on the above-mentioned computers should be able to run on any machine, regardless of the operating system and the CPU used (platform independency). Thus, the Java language was used throughout the software development, which is also a very efficient language for network programming. A detailed description follows for each program developed.

2.1. The RES-Collector

The RES-Collector units are located usually in geographically remote areas, so they must be operationally independent from the RES-Server. In addition, the RES-Collector unit must be able to receive data reliably even in noisy environments, such as the environment of a RES plant. The RES-Collector communicates with the RES data acquisition system via an RF link. The RF transceiver on the RES plant side broadcasts the sampled data, while the RF transceiver on the RES-Collector side forwards the received data to the RES-Collector unit through its RS-232 serial interface. The data are formed in packets starting with the character 's' and ending with the character 'e'. Each packet contains three identical streams of bytes. Each stream consists of as many byte sequences as the total number of measured signals. Each sequence contains a byte corresponding to the measured signal ID number and the bytes corresponding to its value. The values are transmitted three times within the packet, so the RES-Collector is able to detect possible transmission errors and reject the corrupted packets.

The RES-Collector program flowchart is shown in Fig. 7. The program input parameters are: (a) the RES-Collector and RES-Server communication port numbers, (b) the RES-Collector serial port number, (c) the serial port transfer rate and (d) the maximum time allowed for the RES-Collector waiting without receiving data before it restarts, in the case of malfunction. Initially, a buffer for input data storage and a suitable socket for the RES-Server communication are created. The buffer stores a packet of data as defined above. Since the RES-Collector has to operate independently from the RES-Server, the socket created is of ServerSocket type, waiting for connection from a RES-Server socket. When new data are available on the serial port, a new event is triggered and the data input is started. Subsequently, the received characters are transformed to integer and real numbers, corresponding to IDs and values, respectively. The program checks for the integrity of the received data packets according to the following protocol:

- The accepted characters within a packet are: '0'–'9', ' + ', ' − ' and the comma symbol,
- All packets must start with character 's' and end with the character 'e',
- A received number, ID or data value, is accepted only if there are at least two identical occurrences within a packet,
- The IDs received must be within 0 and MAX, where MAX is the total number of signals measured by all RES-Collectors and
- Any packet containing data not fulfilling the above requirements is rejected.

The data transmission to the RES-Server is carried

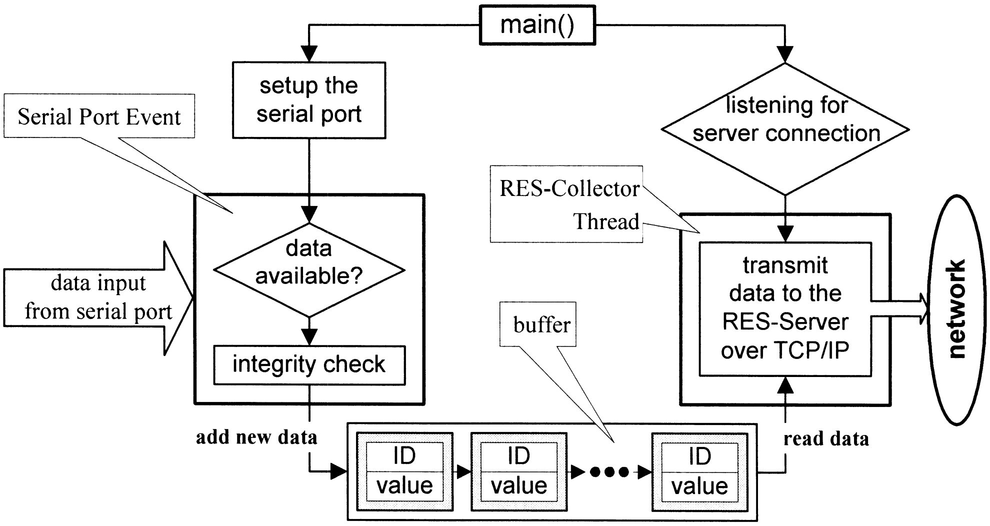


Fig. 7. The RES-Collector program flowchart.

out with a thread (RES-Collector thread in Fig. 7) having a 200 ms repetition time, over an Ethernet network or the Internet, using the TCP/IP protocol. The repetition time selected is a trade-off between CPU loading and system response time.

2.2. The RES-Server

The RES-Server unit collects the data from all available RES-Collectors and transmits them over the Internet to any remote user. All incoming data are labeled with the RES-Collector identification and the date.

The Server program flowchart is illustrated in Fig. 8. The program input parameters are: (a) the number of interconnected RES-Collectors, (b) their identification numbers (IP addresses) and (c) the transmission rate to the RES-Applet. Initially, a buffer storing the incoming data and a new thread for every RES-Collector in operation is created. Each RES-Collector-thread creates a new socket for connection with the corresponding RES-Collector, with a 50 ms repetition time, receives the incoming data and stores them simultaneously (a) in the buffer for transmission over the Internet and (b) in a file on the RES-Server's disk. After settling the interconnection with the RES-Collectors, the program is ready to communicate with any user by means of the RES-Applet. If a user requests connection, a new client socket and a new thread (RES-Applet-thread) are created. The data stored in the buffer are transmitted over the Internet using the TCP/IP protocol in regular intervals, according to the transmission speed set.

2.3. The RES-Applet

The program required to display the measurements on the remote user computer (RES-Applet) is developed according to the Sun Applet format. It is also written in Java language and it is executed on the user computer when the corresponding Web page is invoked. Then, all collected data are displayed in a graphical environment.

The RES-Applet flowchart is shown in Fig. 9. Initially, the graphic interface objects are set up and a client socket is created to establish communication with the RES-Server. The rest of the program, which performs the data import and display, is executed as a thread with a 200 ms repetition time. The repetition time selected is a trade-off between user CPU loading and display refresh rate.

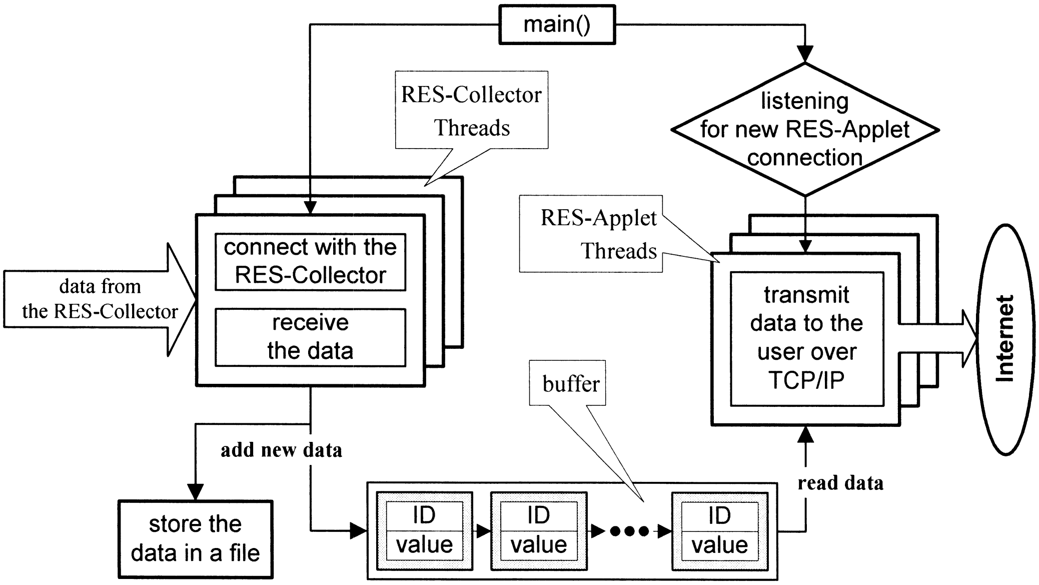


Fig. 8. The RES-Server program flowchart.

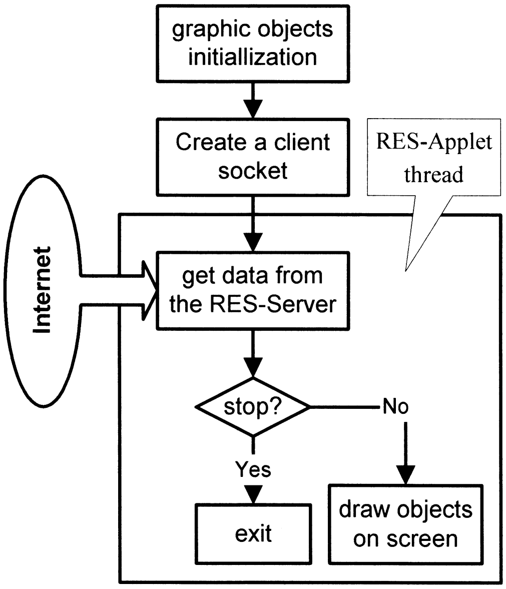


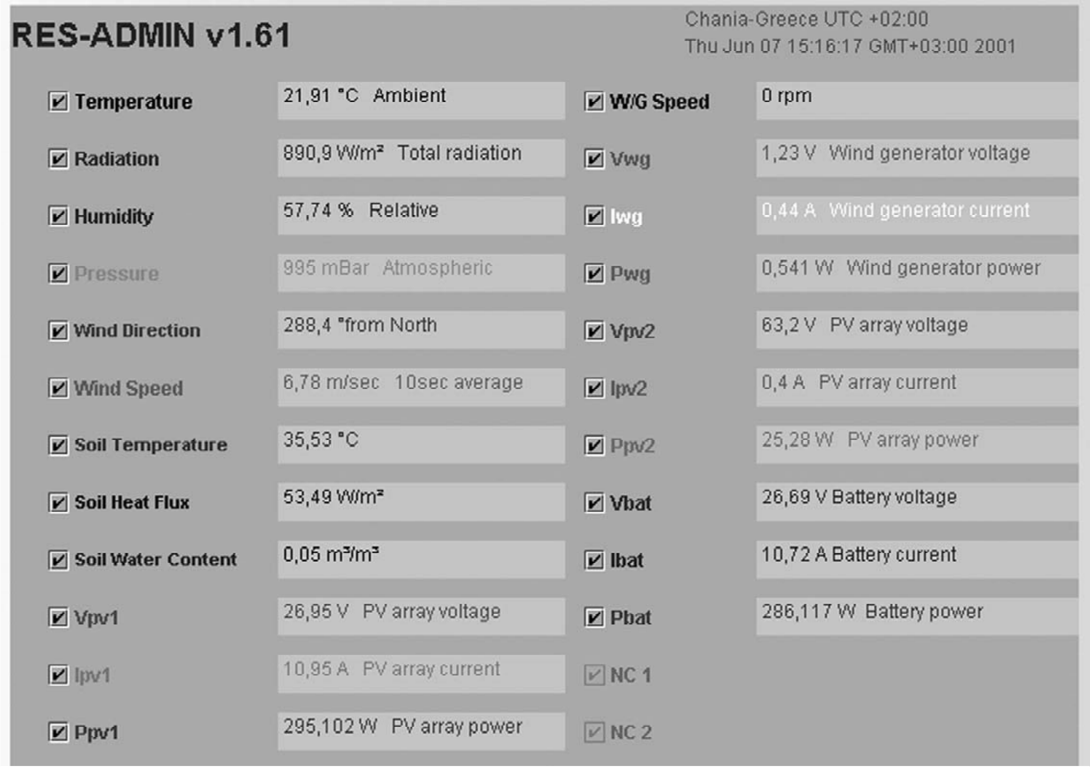
Fig. 9. The RES-Applet program flowchart.

3. An implementation example

A fully functional distributed measurement system based on the above-described architecture is implemented at the Technical University of Crete campus. A detailed description of the system follows.

A number of meteorological and electrical parameters are monitored at a remote RES system, consisting of a wind generator, two photovoltaic arrays, a battery system, battery chargers and inverters, using a suitable laboratory-built data acquisition unit, which communicates with a RES-Server unit by means of an RF link. The parameters measured are: (a) voltage and current of both photovoltaic arrays, (b) the wind generator voltage, current and rotational speed, (c) the batteries voltage and current, (d) wind speed and direction, (e) ambient temperature, humidity, atmospheric pressure and global solar irradiation and (g) temperature, heat flux and water content of the soil. The data acquisition unit transmits a complete set of the above measured data every 1 min. Since the change rate of the measured signals is low, this transmission rate is adequate.

The distance between the RES plant and the RES-



RES-ADMIN v1.61

Chania-Greece UTC +02:00
Thu Jun 07 15:16:17 GMT+03:00 2001

☑ Temperature	21,91 °C Ambient	☑ W/G Speed	0 rpm
☑ Radiation	890,9 W/m^2 Total radiation	☑ Vwg	1,23 V Wind generator voltage
☑ Humidity	57,74 % Relative	☑ Iwg	0,44 A Wind generator current
☑ Pressure	995 mBar Atmospheric	☑ Pwg	0,541 W Wind generator power
☑ Wind Direction	288,4 °from North	☑ Vpv2	63,2 V PV array voltage
☑ Wind Speed	6,78 m/sec 10sec average	☑ Ipv2	0,4 A PV array current
☑ Soil Temperature	35,53 °C	☑ Ppv2	25,28 W PV array power
☑ Soil Heat Flux	53,49 W/m^2	☑ Vbat	26,69 V Battery voltage
☑ Soil Water Content	0,05 m^3/m^3	☑ Ibat	10,72 A Battery current
☑ Vpv1	26,95 V PV array voltage	☑ Pbat	286,117 W Battery power
☑ Ipv1	10,95 A PV array current	☑ NC 1	
☑ Ppv1	295,102 W PV array power	☑ NC 2	

Fig. 10. The RES-Applet interface.

Collector unit is approximately 200 m, covered by the RF link. Both, the RES-Collector and the RES-Server programs are running on a PC with the Windows NT operating system, which also runs the Web server of the laboratory. All the above-mentioned data are available to any user, simply invoking the laboratory Web page (http://www.elci.tuc.gr). The interface appearing on the user display is presented in Fig. 10. The users can select or deselect the measurements they are interested in.

4. Conclusions

The development of a data acquisition system for remote monitoring the operation of a RES plant is analyzed in this paper. It is based on Client/Server architecture and it does not require a physical connection, e.g. through network, serial communication port or standard interface such as the IEEE-488, of the monitored system with the data collection server. This feature is essential in most RES systems since they are usually installed in inaccessible areas.

Furthermore, the data are distributed over the Internet to any remote user. The measurements can be retrieved by means of a properly developed Applet interface by simply visiting the laboratory Web page.

A fully functional implementation of the above described distributed measurement system is operating in our Institution and the data are available over the Internet (http://www.elci.tuc.gr).

The proposed system can be easily extended to perform remote control of the RES system, for users authorized for such a function.

References

[1] D. Grimaldi, L. Nigro, F. Pupo, Java-based distributed measurement systems, IEEE Trans. Instrum. Meas. 47 (1) (1998) 100–103.

[2] M. Abdelrahman, A. Rasheed, A methodology for development of configurable remote access measurement system, ISA Trans. 39 (2000) 441–458.

[3] B. Qiu, H.B. Gooi, Web-based SCADA display systems (WSDS) for access via internet, IEEE Trans. Power Systems 15 (2) (2000) 681–686.

[4] C. Young, W. Juang, M.J. Devaney, Real-time intranet-controlled virtual instrument multiple-circuit power monitoring, IEEE Trans. Instrum. Meas. 49 (3) (2000) 579–584.

[5] M. Bertocco, F. Ferraris, C. Offelli, M. Parvis, A client-server architecture for distributed measurement systems, IEEE Trans. Instrum. Meas. 47 (5) (1998) 1143–1148.

[6] K.B. Lee, R.D. Schneeman, Distributed measurement and control based on the IEEE 1451 smart transducer interface standards, IEEE Trans. Instrum. Meas. 49 (3) (2000) 621–627.

[7] G. Fortino, L. Nigro, Development of virtual data acquisition systems based on multimedia internet working, Comput. Standards Interface. 21 (2000) 429–440.

[8] P. Arpaia, A. Baccigalupi, F. Cennamo, P. Daponte, A measurement laboratory on geographic network for remote test experiments, IEEE Trans. Instrum. Meas. 49 (5) (2000) 992–997.

[9] M. Lundqvist, C. Helmke, H.A. Ossenbrink, ESTI-LOG PV plant monitoring system, Solar Energy Mater. Solar Cells 47 (1997) 289–294.

[10] M. Benghanem, A. Maafi, Data acquisition system for photovoltaic systems performance monitoring, IEEE Trans. Instrum. Meas. 47 (1) (1998) 30–33.

[11] R. Mukaro, X.F. Carelse, A microcontroller-based data acquisition system for solar radiation and environmental monitoring, IEEE Trans. Instrum. Meas. 48 (6) (1999) 1232–1238.