

# Vehicle Tracking Using a Reliable Embedded Data Acquisition System With GPS and GSM

T. KRISHNA KISHORE<sup>†</sup>, T.SASI VARDHAN<sup>††</sup>, N.LAKSHMI NARAYANA<sup>††</sup>

<sup>†</sup> Asst.Prof., Dept. of CSE, ST.ANN'S COLLEGE OF ENGG. & TECH, CHIRALA, PRAKASAM (Dt), A.P, INDIA

<sup>††</sup> Asst.Prof., Dept. of CSE, NIMRA COLLEGE OF ENGG. & TECH, VIJAYAWADA, A.P, INDIA.

<sup>†††</sup> Asst.Prof., Dept. of CSE, ST.ANN'S COLLEGE OF ENGG. & TECH, CHIRALA, PRAKASAM (Dt), A.P, INDIA.

## Abstract

In this paper, we present the principles of a low operational-cost but flexible Internet-based data-acquisition system. The main core of the system is an embedded hardware running a scaled-down version of Linux: a popular choice of operating system for embedded applications. The embedded device communicates through General Packet Radio Service (GPRS), which makes it accessible from anywhere in the world through a web server built into the embedded device. In addition, GPRS provides a bidirectional real-time data transfer allowing interaction. The proposed system eliminates the need for server software and maintenance. A novel approach is introduced to minimize the operational costs while operating with a large amount of data. The system is demonstrated to be suitable for different embedded applications by attaching several real-time modules through appropriate interfaces.

## Key Terms :

GPS, GPRS, GSM, Real Time Processing, Data-acquisition, Embedded System.

## I. INTRODUCTION

DATA-ACQUISITION systems with remote accessibility are in great demand in industry and consumer applications. In some applications, human beings have been replaced by unmanned devices that will acquire data and relay the data back to the base [1]. There are data-acquisition and control devices that will be a substitute for a supervisor in a multisite job operation. A single person can monitor and even interact with the ongoing work from a single base station. An acquisition unit designed to collect data in their simplest form is detailed in [2], which is based on Linux [3], which is a popular choice for embedded PC systems. A similar system in [4] provides data acquisition with no concern for remote access. Data collection for post processing on a vehicle's position for an advanced traffic survey is discussed in [5]. Some applications adding remote accessibility are detailed in [6] and [7], which are built to collect and send data through a modem to a server. Although these are well-built systems that

serve the purpose for a specific task, the user cannot interact with the system. Another unidirectional data transfer is presented in [8], which uses the Global System for Mobile Communications (GSM): a popular wireless choice for connectivity between the data-acquisition units and clients. There are also several systems that allow data to be remotely accessed. As a solution to wireless data collection through the Internet, General Packet Radio Service (GPRS) is a popular choice in several applications. A surveillance system based on GPRS is presented in [9]. A recent work [10] has represented a GPRS solution to the data-acquisition problem for remote. A distributed system capable of road vehicle locating, monitoring, and telemetering with GPRS is presented in [11]. A long-distance data-collection system for the Earth tide gravimeter, collecting information on temperature, humidity, atmospheric pressure, etc., is designed with GPRS using a hardcoded static Internet Protocol (IP) address [12]. These systems use GPRS without concerns about minimizing the cost of data transfer. Similar types of Internet-based systems, such as those in [13]–[16], are designed to gather a bulk of data before serving them upon request. In these applications, data are compiled in a central server and are then served to the clients via the Internet. The client framework is in a central server and has all the applications. A person that needs to access any data must first access the server. An indirect access to the data-acquisition unit makes the system unattractive for real-time control applications, where direct interaction with the system may be required. The need to maintain an additional server will also increase the setup costs and the costs to maintain the acquisition systems, such as regular maintenance costs, system updates, etc. Therefore, the central server has to be eliminated for a real time system. The closest to this idea is published in [17]. In this system, a reliable bidirectional Point-to-Point Protocol (PPP) link for real-time control and surveillance via a GSM network is formed. However, there is still no effort to minimize the operational costs (including the costs to transfer a large

amount of data). In addition, this system is based on an industrial PC, thus making it an expensive solution. Interaction with the embedded unit is also an important issue. In [18], an embedded PC card placed on the Internet allows limited interaction through commands sent through Transmission Control Protocol/IP (TCP/IP) and User Datagram Protocol. In this paper, we propose a GPRS-based portable low-cost data-acquisition system, which can establish a reliable bidirectional connection for data-acquisition. The proposed system uniquely reduces the costs occurring from frequently requested data and eliminates the need for a well-established server. The system uses a dummy server for static information, thus optimizing the transfer of large data. The user can directly log in and interact with the embedded device in real time without the need to maintain an additional server. The system is modularly built, allowing different modules to be added. In addition, it is

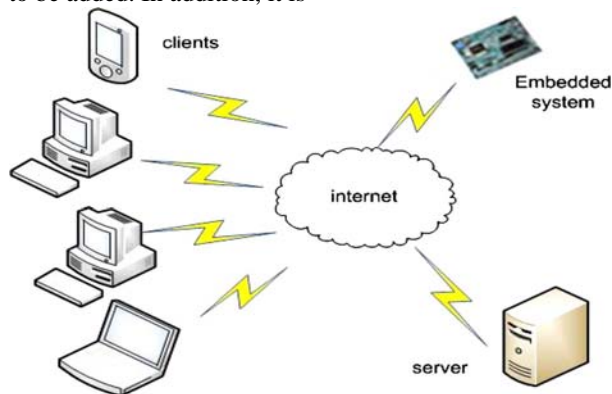


Fig. 1. General diagram of a data-acquisition and control system

flexible to accommodate a wide range of measurement devices with appropriate interfaces. In Section II, we will introduce the details of the aforementioned system. In Section III, we will provide an example that will introduce some of the capabilities of the system using a collection of modules. In this paper, a camera, a Global Positioning System (GPS), and a temperature chip are connected to the embedded system to demonstrate its operation. The operational features are presented, which are needed to be considered for practical applications. Section IV presents the conclusion.

## II. INTERACTIVE DATA-ACQUISITION SYSTEM

The general principles of Internet-based control systems have been modeled in [19]. Interactive Internet-based systems provide a way to monitor and adjust using standard web browsers and a PC. The target systems can be monitored and controlled independent from the

location and the platform since standard web browsers can be used on the client side. A typical data-acquisition system is made up of three components connected to each other via the Internet, as shown in Fig. 1. The data-acquisition system needs to relay the acquired information to the requesting clients. The clients also need to send commands. If necessary, this is implemented through a server, and then, an enormous amount of data transfer time would be consumed. Thus, alternative methods need to be explored.

### A. Establishing a Direct Communication Link Between the Client and the Embedded Device

GSM and GPRS [20] are developed for cellular mobile communication. A GPRS connection with unlimited duration of connectivity is charged only for the data package transfers and adopted in several mobile remote control/access systems [13], [14], [16]. GPRS becomes a cost-effective solution only if the data transfers can be optimized. Once a GPRS connection has been established, queried data can be relayed to the client via a central server [13]–[16]. Using a central server to relay the acquired data has some disadvantages. First, a central server needs a client interface framework. An additional data transfer corresponds to time delays before the data are made available to the client. In addition, since the server acts as a relay, no direct bidirectional communication between the client and the embedded system can be established.

This makes the system unsuitable for real-time control applications. The basic idea behind real-time processing is that the embedded system is expected to respond to the queries in time. Real time should be fast enough in the context in which the system is operating and reliable as well. Real-time system correctness depends not only on the correctness of the logical result of the computation but also on the result delivery time [21]. This method also increases the data transfer cost as the number of clients increases due to the access amount of data transfers via GPRS. Direct communication, on the other hand, enables access to only relevant information in the embedded system by pre processing the data. The embedded system should also handle the web services. This eliminates the need for a central server and reduces the amount of data sent from the remote unit since only the queried data will be transferred. In the proposed system, the GPRS architecture and protocols are compliant with [20]. This system is configured to be virtually online at all times in a GSM network. An admin script is executed after the boot of the operating system, initiating the GPRS connection software module. A PPP connection is established by a GPRS modem that works at 900/1800/1900 MHz operating frequencies. A PPP daemon (PPPD) is used to manage the PPP network

connections between the client and the embedded module. The PPPD is responsible for setting up the GPRS parameters, such as the connection speed and compression. To directly access an embedded system, the IP address of the embedded device should be made available to the client side. There are two choices available. A static (hard-coded) IP could be used, or the remote device should initiate a connection by reporting its IP. This choice is quite straightforward and simple. Although the usage cost remains unchanged, it requires a static IP setup by the service provider and involves monthly recurring costs. The static IP is preferred for its simplicity in designing a system; however, its overhead may be impractical. The other choice is to use a dynamic IP assigned through a Dynamic Host Configuration Protocol (DHCP) server of the GSM provider for every connection established. However, this IP needs to be known by any client requesting an access to the embedded server. One solution is to broadcast this IP to a dummy FTP server (where the bulky static information such as image data is also kept). The FTP server is a dummy server and does not require regular software updates or maintenance. The folder structure of the FTP server is shown in Fig. 2. A script on the embedded device is configured to update its IP address on the FTP server in Hypertext Meta-Language as an index.htm file, under a folder uniquely named by its hostname. This script simply parses the current IP for that embedded device and sends an htm file with the IP information of the embedded device to the FTP server. Once this file is in place, a direct connection can be established with the desired embedded device by a simple query. An example embedded system, named mozart (Fig. 2), can be queried from the FTP server by a simple command. The web browser processes the (index.htm) file in the specified folder as default; therefore, a file name is not needed for referencing.

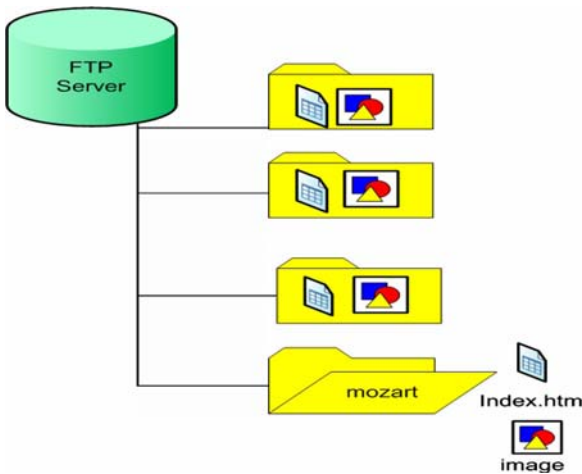


Fig. 2. A script on the embedded device

This process is illustrated in Fig. 3. The DHCP approach is more flexible and works better compared with the static approach as a cost-effective solution, despite the necessity for a script running on the embedded server, one-time broadcasting its IP to the FTP server. The hypertext file placed on the FTP server by the embedded system and queried by the client is shown in Fig. 4. With this mechanism in place, the embedded system updates its IP information on the FTP server upon every reboot, which causes an IP refresh from the GSM service-provider.

**B. Data Management in the System**

The Internet server is used to decrease the management costs by sending all the pictures (logo, picture, bar graphics, etc.) to the client through a server on the Internet. Text data such as coordinates, temperature, and altitude are served from the embedded system. If bulky data are going to be sent, the embedded module is set to send the image only once via GPRS and placed on an FTP server. This approach eliminates the transfer of large data through GPRS more than once, thus reducing the transfer costs, particularly if more than one client is involved or multiple requests to the same data are needed, as shown in Fig. 5. A user interface, which is brought up upon establishing a direct connection, has links to the Common Gateway Interface (CGI) and Bourne Again Shell (BASH) script files executed on the embedded system. The code is compiled into the CGI format to be installed in the embedded board through a cross-compiler platform [22]. BASH scripts are directly triggered by the applications.

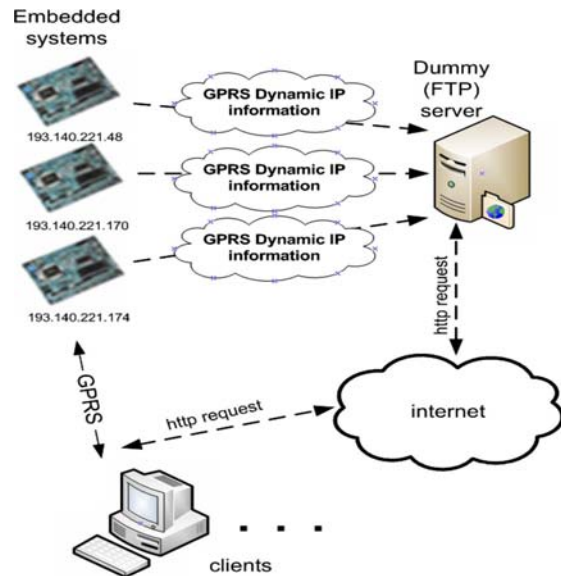


Fig. 3. IP address lookup for the stored IP.

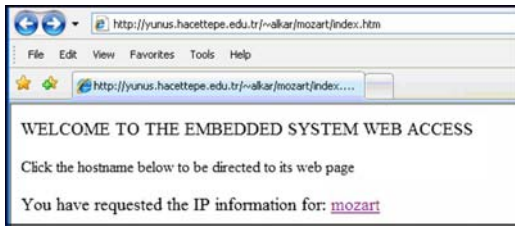


Fig. 4. Index.htm on the FTP server as viewed by the client.

C. Hardware

The embedded system used in this work is an X86-based standalone unit with four serial ports and a parallel port with 16-MB onboard removable Flash memory, as shown in Fig. 6.

One of the serial ports is used in the application design stage for debugging purposes, and this port is designed to host more devices with a multiplexer unit. The other serial ports used by the modules are used to test the system functionality. The acquisition units on the device can be varied with no limitation on their functionality and can be added by using appropriate interfaces.

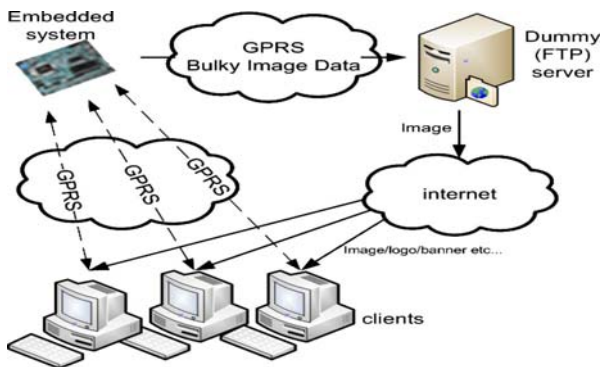


Fig. 5. Data management in the proposed system

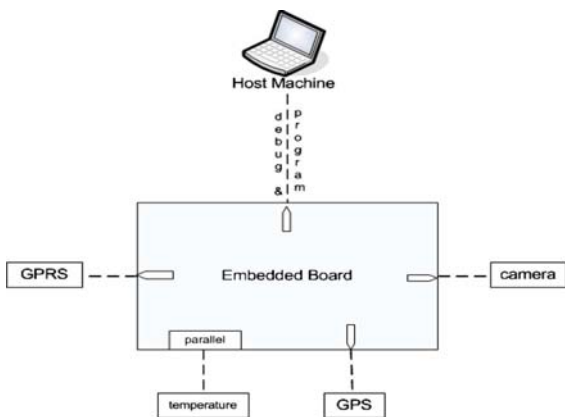


Fig. 6. Block diagram of the embedded system with sample devices attached.

D. Software and Operating System Choice

The Linux 2.4 kernel series [23] with TCP/IP stack included has been chosen as the operating system for the embedded board. Only the bare minimum is installed, including the basics such as console tty, serial ports, kernel side of the PPPD, and support for memory and math emulation. The running kernel is around 1MB of code built into a Flash memory. A scaled-down version of Linux has been used to reduce the memory footprint and the complexity.

The software running on the embedded system at the highest level is named the manager code, which will be explained in Section III with a sample implementation. In the design, the manager code controls the execution of other applications and is triggered once all the components of the operating system are up and running. The flowchart of this code is given in Fig. 7, representing a sample operation of the GPS unit that checks the execution if the speed limit has been exceeded. The periodic operations and routine tasks are organized by a manager code. If a new data-access application is considered in future developments, its program can easily be added to the manager code as a periodic operation.

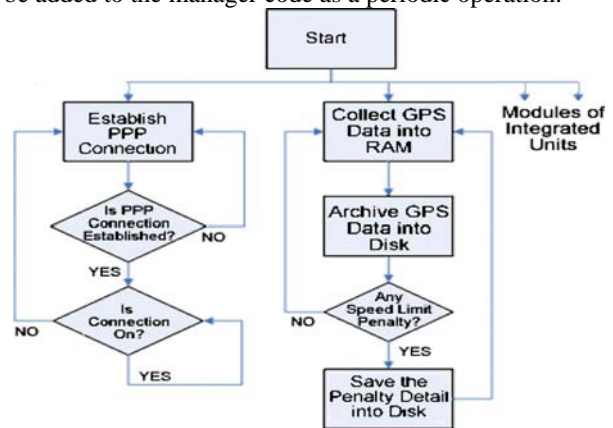


Fig. 7. Manager code operational principles.

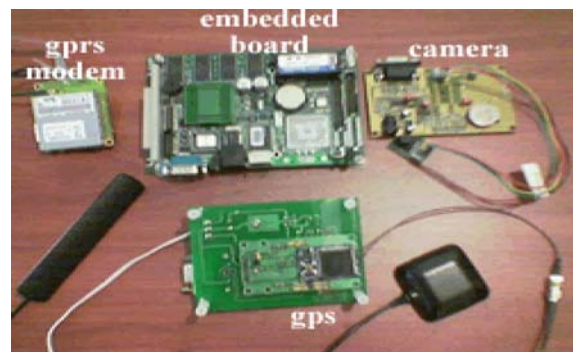


Fig. 8. Components of the embedded system.

### III. SAMPLE APPLICATION

A camera, a temperature sensor, and a GPS are integrated into the embedded board to form a sample application, as shown in Fig. 8. These units and their interaction with the embedded board are briefly explained in the following discussion. In addition, the time delays at each operation stage are discussed to demonstrate the effectiveness of the proposed method. A time delay of 0.57 s, on average, is needed for any control command to be sent. This delay is related to the GPRS service of the GSM service provider.

TABLE 1 : TOTAL CONNECTION TIME TO RECEIVE A 16-kB PICTURE THROUGH AN FTP SERVER AND DIRECT ACCESS

# of clients at the same time	$\Sigma$ Duration (sec) to receive pictures	
	FTP server	Direct Access
1	21.77	21.77
2	21.77	40.14
3	21.77	58.51
4	21.77	76.88

For this sample application, a complementary metal-oxide-semiconductor camera with a built-in JPEG codec controller chip has been chosen. The camera acquires bulk image data; therefore, it is a good module to demonstrate the effectiveness of the system. It compresses and transfers the image from the camera to the serial port. The communication with the camera is established over an RS232 communication protocol using an asynchronous package transfer method. Before taking a snapshot, the camera is synchronized by sending an appropriate number of synch data packages. After the synchronization, both the embedded board and the camera wait until they receive an acknowledgement from the other side before sending another request or data. This protocol is executed in an average of 3.4 s for each picture, which can be considered as an adequate rate for most applications. Here, the bottleneck is the camera; hence, the speed of data transfer can further be improved by using a camera with a faster sampling rate. The client initiates the camera control script, which eventually takes a snapshot. The embedded board receives the data from the camera port then stores them into the Flash memory externally added onto the embedded unit. The available data storage in 16-MB Flash memory is 6.6 MB, which is suitable for over 420 pictures. The picture is uploaded (a 16-kB JPEG picture is transferred in an average of 17.8 s) to a dummy FTP server, as described in Section II-A. The server on the Internet is not maintained and only used for storage space. Since, for our application, a history is not required to be kept, the client(s) accessing the picture download(s) the most recent snapshot from the FTP server. All the

queries to visualize the current picture are automatically relayed to the FTP server instead of the embedded module, thus reducing the operational costs. The duration comparison of using a dummy server with respect to direct access is shown in Table I.

The scenario of serving a single picture to a maximum of four clients is shown in Fig. 9. The x-axis represents the number of client accesses to a picture. For the camera application, the client can take a snapshot and visualize the picture on the screen (an average of  $0.57 + 3.4 + 17.8 = 21.77$  s for a 16-kB picture) in less than 22 s, including the time delay of command execution, camera operation, and picture upload duration. Although this may seem like a large delay, it may be improved with a faster and more expensive camera. The transfer of text data takes an average of less than 1 s ( $\sim 1$  kB).

### IV. CONCLUSION

In this application, a low-cost, Internet-based data acquisition and control system has been designed and implemented that should find interest from researchers. The application possibilities are virtually unlimited by attaching modules with appropriate interfaces, although the usage of the system is demonstrated with only a few sample devices.

Compared with other applications, this system has advantages in terms of allowing direct bidirectional communication and reducing overhead, which can be vitally important for some real-time applications. The operational costs have been reduced by relinquishing the storage of large data to an FTP server on the Internet. The system is designed to support both static and dynamic IPs. A method to distribute the IP information has been developed. This cost-minimization effort is a big concern for mobile systems using wireless communication methods and has not been discussed before. The overall cost advantage of the system in terms of the components used makes it an attractive choice for data-acquisition applications. The power demand of the device is still in the process of being improved by putting the attached devices into sleep mode at times when they are not in use to conserve power.

### REFERENCES

- [1] C. E. Lin, C.-W. Hsu, Y.-S. Lee, and C.C.Li, "Verification of unmanned air vehicle flight control and surveillance using mobile communication," *J. Aerosp. Comput. Inf. Commun.*, vol. 1, no. 4, pp. 189–197, Apr. 2004.
- [2] K. Jacker and J. Mckinney, "TkDAS—A data acquisition system using RTLinux, COMEDI, and Tcl/Tk," in *Proc. Third Real-Time Linux Workshop*, 2001. [Online]. Available: The Real Time Linux Foundation:

- <http://www.realtimelinuxfoundation.org/events/rtlws-2001/papers.html>
- [3] E. Siever, A. Weber, S. Figgins, and R. Love, *Linux in a Nutshell*. Sebastopol, CA: O'Reilly, 2005.
- [4] Q. Zhou, W. Wu, and Y. Ma, "The embedded data acquisition system for Mössbauer spectrum," in Proc. Third Real-Time Linux Workshop Embedded Linux Expo Conf. Real-Time Embedded Comput. Conf., Milan, Italy, Nov. 2001, pp. 26–29.
- [5] J. E. Marca, C. R. Rindt, M. McNally, and S. T. Doherty, "A GPS enhanced in-vehicle extensible data collection unit," Inst. Transp. Studies, Univ. California, Irvine, CA, Uci-Its-As-Wp-00-9, 2000.
- [6] W. Kattanek, A. Schreiber, and M. Götze, "A flexible and cost-effective open system platform for smart wireless communication devices," in Proc. ISCE, 2002.
- [7] J. E. Marca, C. R. Rindt, and M. G. McNally, "The tracer data collection system: Implementation and operational experience," Inst. Transp. Studies, Univ. California, Irvine, CA, Uci-Its-As-Wp-02-2, 2002.
- [8] E. Bekiroglu and N. Daldal, "Remote control of an ultrasonic motor by using a GSM mobile phone," *Sens. Actuators A, Phys.*, vol. 120, no. 2, pp. 536–542, May 17, 2005.
- [9] C. E. Lin and C.-C. Li, "A real-time GPRS surveillance system using the embedded system," *J. Aerosp. Comput. Inf. Commun.*, vol. 1, no. 1, pp. 44–59, Jan. 2004.
- [10] C. Xiaorong, S. Zhan, and G. Zhenhua, "Research on remote data acquisition system based on GPRS," in Proc. 8th ICEMI, 2007, pp. 2-20–2-23.
- [11] M. A. Al-Tae, O. B. Khader, and N. A. Al-Saber, "Remote monitoring of vehicle diagnostics and location using a smart box with Global Positioning System and General Packet Radio Service," in Proc. IEEE/ACS AICCSA, May 13–16, 2007, pp. 385–388.
- [12] C. Zhang, J. Ge, H. Yu, and X. Zhang, "ET gravimeter data collecting system based on GPRS," in Proc. 8th ICEMI, Jul. 18–Aug. 16, 2007, pp. 2-86–2-92.
- [13] A. Sang, H. Lin, and C. E. Y. Z. Goua, "Wireless Internet-based measurement architecture for air quality monitoring," in Proc. 21st IEEE IMTC, May 18–20, 2004, vol. 3, pp. 1901–1906.
- [14] G. Zhenyu and J. C. Moulder, "An Internet based telemedicine system," in Proc. IEEE EMBS Int. Conf. Inf. Technol. Appl. Biomed., 2000, pp. 99–103.
- [15] J. Dong and H. H. Zhu, "Mobile ECG detector through GPRS/Internet," in Proc. 17th IEEE Symp. CBMS, Jun. 24–25, 2004, pp. 485–489.
- [16] P. Wang, J.-G. Wang, X.-B. Shi, and W. He, "The research of telemedicine system based on embedded computer," in Proc. 27th IEEE Annu. Conf. Eng. Med. Biol., Shanghai, China, Sep. 1–4, 2005, pp. 114–117.
- [17] C. E. Lin, C.-C. Li, A.-S. Hou, and C.-C. Wu, "A real-time remote control architecture using mobile communication," *IEEE Trans. Instrum. Meas.*, vol. 52, no. 4, pp. 997–1003, Aug. 2003.
- [18] T. Motylewski, "The industrial data-acquisition system with embedded Rt-Linux and network server technology," in Proc. Third Real-Time Linux Workshop, 2001. [Online]. Available: The Real Time Linux Foundation: <http://www.realtimelinuxfoundation.org/events/rtlws-2001/papers.html>
- [19] S. H. Yang, L. S. Tan, and X. Chen, "Requirements specification and architecture design for Internet-based control systems," in Proc. Int. Comput. Softw. Appl. Conf., Dev. Redev., 2002, pp. 75–80.
- [20] C. Bettstetter, H.-J. Vögel, and J. Eberspächer, "GSM phase 2+ General Packet Radio Service GPRS: Architecture, protocols, and air interface," *IEEE Commun. Surveys Tuts.*, vol. 2, no. 4, pp. 2–14, Third Quarter 1999.
- [21] RTOS Evaluation Project, "What makes a good RTOS," *Dedicated Systems Experts*, 2001. [Online]. Available: <http://www.dedicatedsystems.com>
- [22] K. Yaghmour, *Building Embedded Linux Systems*. Sebastopol, CA: O'Reilly, 2003.
- [23] C. Hallinan, *Embedded Linux Primer: A Practical Real-World Approach*. Englewood Cliffs, NJ: Prentice-Hall, 2006.