REMOTE TEMPERATURE RECORDING USING BLUETOOTH TECHNOLOGY

Marian ALEXANDRU     Mihai ROMANCA     Petre OGRUTAN
Transilvania University, 1 – 3 Politehnicii Str Brasov, Romania
Tel/Fax +40268478705, alexandrum@vega.unitbv.ro; romanca@unitbv.ro

Abstract: The past decade has seen significant advancement in the field of consumer electronics. Various “intelligent” appliances such as cellular phones, air-conditioners, home security devices, indoors localization, etc. are set to realize the concept of a smart home or building. A wireless home network that does not incur additional costs of wiring would be desirable. Bluetooth technology, which has emerged in late 1990s, is an ideal solution for this purpose. This paper describes an application of Bluetooth technology in home or industrial automation and networking environment. It proposes a network, which contains a remote temperature sensor connected to a microcontroller, which controls, also, the Bluetooth module transmission and a computer having internet access, connected via USB to a Bluetooth module. In this way, the sensor data are collected and processed by the microcontroller, transmitted via Bluetooth to a fixed or mobile computer and published on a web page in order to be accessible for a remote user.

Key words: Bluetooth, smart building, temperature recording

I. INTRODUCTION

Wireless technology for the temperature recording is proposed for mobility, low cost, low power, small size and communication over short distances in a home or industrial environment. ISM band communication technologies, such as infrared, Zigbee and Bluetooth, are of particular interest in these features. The conclusions from the investigation were: The drawback of infrared technology is the requirement of a line of sight between sender and receiver. This makes infrared a reluctant choice for the transmission medium in a sensor network scenario, particularly in the home and industrial environments, because it is very easy for the transmission path to be blocked. Zigbee is an ideal technology for this application; however, it is not yet available for real application, due to the relatively short time since the introduction of the Zigbee and the standardization. Bluetooth has been used in a wide range and the technology is available in the market. These encourage us to introduce it to remote temperature recording.

Bluetooth is an open standard for short-range digital radio. It is designed to operate in the unlicensed ISM band. It has been developed to set-up “Pico” networks. Bluetooth uses a master/slave-based MAC protocol and enables low power consumption and short-range wireless connection between various electronic devices. Bluetooth is intended to replace the cables connecting portable and/or fixed electronic devices. The technology also offers wireless access to LANs, PSTN, mobile phone networks and the Internet. Wireless Internet access using Bluetooth is becoming popular. Applications in accessing the Internet have been investigated in experiments by Tan and Soh [1] for home control systems. A home automation system, using Bluetooth, has been experimentally built up with Sriskanthan et al [2]. Integration of Bluetooth and mobile products has been discussed in a study by Kirby [3]. It is already possible for Bluetooth to extend its application scope to 3G wireless communication systems. The possibility of an industrial use for Bluetooth in data collection has been discussed in a study by Anderson [4]. Bluetooth using data transmission from inside vacuum chamber measurements [5] is an actual example of industrial usage. Also, the authors tested the interferences between Bluetooth and other wireless technologies [6], susceptible to be present in a home or industrial environment.

II. THE PROTOTYPE

The prototype of remote temperature recording (RTR) is designed for application to the area of home and industrial temperature monitoring. RTR has functions of measurements of air temperatures, A/D conversion, measurement calculations and wireless data transmission. The firmware and hardware for RTR have been developed in house. The following sections give details of RTR.

A. The RTR hardware

RTR hardware consists of a microcontroller, the Bluetooth module (BTM), and the temperature sensor, as shown in Fig.1. The sensor measures the air temperature in the tested environment. The microcontroller, with embedded software, does measurements and calculations. The microcontroller commands BTM activities, such as inquiry, connection and reset. The microcontroller manipulates the sensor data transmission to BTM. Then wireless data
Transmission is carried out by BTM. The system realizes the measurement of the temperature and has been conceived as an ensemble of two sub-systems, which interact using the Bluetooth communication protocol: the main server and the local system of data acquisition.

For distance communication, the system connects to one of the terminals and interposes the communication with a second terminal implicated in the application. The local acquisition of the temperature signal is accomplished using an ATmega8 with micro-controller. Also, using the created firmware component, the system can control the data acquisition from the sensor, the received data processing, the data sending to the LMX9820A Bluetooth module through USART and the distance communication between the above mentioned terminals using Bluetooth protocol.

The communication between the acquisition board and the PC is succeeded using the Bluetooth module LMX9820A, which practically emulates a serial communication. The data is acquired, processed, and sent to the PC in a time interval, which is pre-established. The computer (PC or portable) has been incorporated with a Bluetooth 2.0 adaptor connected to the USB interface. The received data are registered to a folder, from where they are collected by the Apache server and inserted into a MySQL data base using the code lines in PHP language. Thus, the information is collected and saved in the data base of the server and can be accessed by a web browser, from any computer connected to the Internet. The temperature is updated on the Internet page in a time interval, and the previous values are stored in the data base. The last n values can be seen; the n dimension can be set by the user.

The Figure 1 shows the block diagram of the hardware system implemented.

For temperature measurement the LM335Z sensor from the National Semiconductor is used, supplied to 5V. Also, a semi adjustable potentiometer of 10KΩ for realizing the sensor calibration is used, thus the sensor voltage drop, at the temperature of 25°C is of 2.98V, and the variation of the voltage with the temperature is linear and of approximately 10mV/°C. The reading of this voltage is made through the ADC5 port of the ATmega8.

The National Semiconductor LMX9820A Bluetooth is a SPP (Serial Port Profile) module with an included chip antenna, having UART interface, which, practically emulates a data serial transmission – the data received on the
UART is sent further also on the UART interface.

This circuit works together with other Bluetooth modules that support the same PPS profile or can connect through the UART serial interface to a processor or direct to the system, depending of the application. Through an external processor or host (personal computer) all the available profile applications could be set to the PPS profile, for example: Dial Up Networking, Fax, LAN Access. LMX9820A can work with an independent slave - module. A master module can support up to 3 slaves.

To manage the connection, it is necessary to know whether the adaptor is compatible with SPP profile. The steps that are made are the following: the software for the UART serial interface to a processor or direct to the PC’s USB and the searching for Bluetooth devices is started; when the LMX9820A is found its name will appear as “Serial Port Device”; right click on the device name that was found and choose the option pairing; the PIN code of the device is inserted – 0000 and OK is pressed. The adaptor’s green LED is turned on, to show that the steps have been made correct.

For sending data, the Windows HyperTerminal is used, set on the COM port and having the following parameters: 8 data bits, 1 stop bit, no parity, hardware protocol, 9600 kbps.

B. The RTR software

The source code has been written in ‘C’ language, using the ICC AVR program.

The bluetooth_online.c contains the functions, which initializes the microcontroller, acquires the sensor data, processes this data and sends them to the central server using the Bluetooth principle.

When the application starts, first of all the microcontroller ports initiate → port_init (). UART interface → uart0_init (), the ADC converter → adc_init(). These procedures are called in the init_devices () procedure.

Then, the acquisition process of the data begins in the conditions that initially, the ADC data entry port was set: the conversion is started; wait until the conversion has ended and the obtained value is stored in the ADC registry; the value from the registry is taken and stored in the internal variable val_adc

Next is the temperature computing, which is made by the function calculez_temp(int); following this call, the temperature value will be found in the int temp variable. After the itoa(buffer, temp, 10) function call, the value from temp is transformed into a line of characters which are stored in the unsigned char buffer[7]. The third parameter of the function – 10, represents the base in which the information from temp is converted.

The characters from buffer are sent one by one to a serial interface, in the background of a for loop, using the function call afisare_caracter(unsigned char).

All the applications described above except the initialization are made every two minutes, this being realized with the function call asteapta_sec(120);

For the server application to work, the installation of the Windows XP compatible XAMPP 32.exe was necessary. This includes the following useful applications for the temperature measurement module development: the Apache server over the PHP language – decodes the PHP language and transforms it into an HTML code which can be displayed by the browsers; My SQL Data Base – the server data base application used for the measured values storing; the PHP language – where the server executed code is written. To write the code the Zend 3.1 editor has been used.

The files that contain the application code are stored in the /xampp/htdocs/artactanap/ directory.

For the server to work and the application to function in optimal parameters, it is necessary that in the XAMPP command panel, the Apache and My SQL DB to be checked and work, exactly as it’s shown in the Figure 2 below:

At the web address /localhost/phpmyadmin/ a new database is created; by pressing the “Import” button the file which includes the created database is brought. The ‘go’ button is pressed, a web browser is started and localhost/artactanap/ is typed. At this moment, the created database has been loaded and the application must work.

For the application to be more reliable, it’s preferred that the server to have a static IP. Thus, the web page can be accessed from anywhere, without accessing restrictions, only user’s right. In the case that the server has a dynamic IP, every time it’s restarted or when the Internet connection is interrupted, it will get another internal IP. This new IP has to be implemented into the application, so it can be accessed.

C. The code description

The application’s code contains the following files: comodici.sql - represents the database of the application; celle.page - when the server administrator modifies, at the address where the application is resident, the way to the file from where the data is taken, this program is called and the new way it’s automatically stored in the database; the new way is displayed; index.php - represents the first page which is loaded on the web, it contains all the obligatory tags for an HTML page. In this file the time interval in which the page displays a new temperature value is set. It also contains all the static elements of the page: the author name, title, etc.; main.php - represents the central page of the application and the dynamic elements of the web page: pictures, current temperature; ultimele_inregistrari.page – it
does a database interrogation and displays the last \( n \) records. In this file the number of the page displayed registrations can be modified; verificare.do – it’s a program that works in the background, reading (at the time interval specified by the index.php file) from the data base the way to the file from where the data is collected, finding this file, reading its contents and erasing it. In case a wrong way in the database is found, an error message is displayed.

D. The user interface

The user interface of the temperature measuring application for a monitorised environment using Bluetooth technology can be accessed from the Internet. The code is written in PHP, which allows dynamic element visualization, database accessing, etc. On the start page of the application it is shown the current temperature.

The page with the last registrations, which shows the last \( n \) temperature values, and also the date and the hour: is presented in the Figure 3.

III. CONCLUSIONS

The application is easy to use and it’s available from a monitoring system situated at a certain distance. The realized user interface has a simple structure and it can be modified easily to display data acquired from numerous sensors. For accessing it, we need only an Internet connection and a web browser. The demonstrative system can be extended for measuring different parameters of interest, like: humidity, gas concentration, presence, etc. SMS, faxes and e-mails can be transmitted to the users in case the system detected parameters pass certain limits.

REFERENCES