Design and Investigation of a WEB Based Temperature Sensor

Angel Iliev Pashev and Peter Ivanov Yakimov

Abstract – **Currently, a great number of users have access to the World Wide Web (WEB). This gives the users access to various information, everywhere at any time. Such information can be temperature. Often it is necessary to distantly monitor the temperature in a room, warehouse or storage area. Using a microcontroller with embedded Ethernet controller and the well known TCP/IP and UDP protocols the monitoring is easy and accessible.**

Keywords – **Ethernet, Protocols, WEB, Temperature**

I. Introduction

Modern ways for measuring temperature are based on smart digital or analog sensors. Most of the digital sensors have standard output on I^2C, SPI or other interface. The analog sensors usually have analog voltage output. The temperature sensors also have good stability and linearity in the range between -80 to +150 degree Celsius. This range makes them practical for measuring ambient temperature. For temperatures in wider range, a thermal couple with a good scheme for linearization and amplifier is often used.

Acquiring a connection to the Internet can be implemented with Ethernet controller connected to or embedded in a microcontroller. Controller of this type has some of the well know producers such as Freescale, ATMEL, Microchip, etc. Most of the vendors ship with the hardware ready to use TCP/IP, UDP, HTTP protocols enabling the users to focus more on the data processing and visualization.

Integration of an analog or digital sensor or actuator element, a processing unit, and a communication interface is called smart transducer in the IEEE 1451 – Networked Smart Transducer Interface Standard. A smart transducer comprises, a hardware or software device consisting of a small compact unit containing a sensor or actuator element, a microcontroller, a communication controller and the associated software form signal conditioning calibration, diagnostics and communication.

Based on this premise a smart transducer model is shown in Fig. 1. It consists of four parts: transducers (sensors and actuators), signal conditioning and data conversation, application processor, and network communication. The analog output of the sensor is conditioned and scaled (amplified), then converted to a digital format by an A/D converter. The digitized sensor signal is processed by a microprocessor using a digital application control algorithm. The measured or calculated parameters can be passed on the host or monitoring system in a network by means of network communication protocols. In a reverse manner, an actuation command send form a host via the network can be used to control an actuator.

A. Pashev is with the Faculty of Electronic Engineering and Technologies, Technical University - Sofia, 8 Kliment Ohridski blvd., 1000 Sofia, Bulgaria, e-mail: angel_pashev@smartcom.bg
P. Yakimov is with the Department of Electronics and Electronics Technologies, Faculty of Electronic Engineering and Technologies, Technical University - Sofia, 8 Kliment Ohridski blvd., 1000 Sofia, Bulgaria, e-mail: pij@tu-sofia.bg

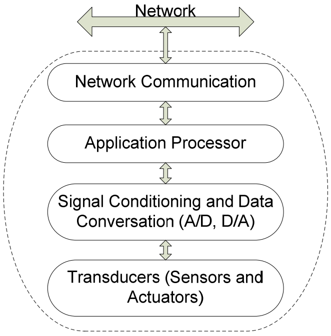

Fig.1. Smart Transducer model

II. Description of the Temperature Measuring System

The system is based on TCP/IP layer network model, shown on Fig. 2. When the designated receiver of the packet is reached (the MAC address in the ARP table match for the Ethernet protocol), the header part of the frame is removed and then the packet is send to the next level. On the second is the Internet Protocol (IP), which checks the IP address and if it matches – its packet header is removed. The packet is then moved to the upper level protocols (TCP-Transmission Control Protocol or UDP – User Datagram Protocol) and after that to the application layer. When sending a packet, each layer adds its header part and sends the packet to the lower levels. When the packet reaches level 1 it is transmitted over the physical network.

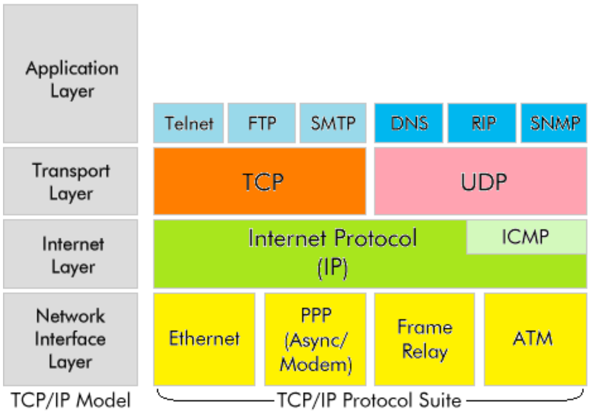

Fig. 2. TCP/IP model layers

The system uses analog sensor directly connected to the analog-to-digital converter (ADC) input of the MC9S12NE64 microcontroller. The schematic is shown on Fig. 3

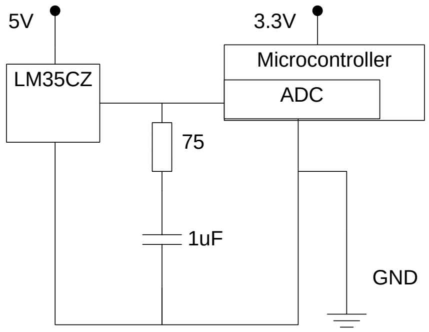

Fig. 3. Sensor's electric circuit

The power supply circuit provided in documentation of the controller [6] is used. It provides 3.3V power supply for the microcontroller and reference voltage for the ADC. However the analog temperature sensor requires higher 5V power supply. For that purpose a 7805 voltage regulator is added.

The LM35CZ sensor [4] is used only for positive temperatures between 0-150 ^{0}C (for ambient temperature in a living room for example). The controller has 10bit ADC which according to the reference voltage 3.3V of the controller has sensitivity of 3.22mV/LSB. The sensitivity of the sensor is 10mV/^{0}C which is 3.1 steps on the ADC or approximately 0.3^{0}C/LSB. Full block scheme of the device is given on Fig. 4.

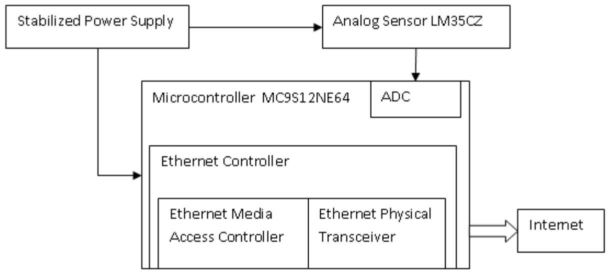

Fig. 4. Measuring system block diagram

The general algorithm of the software is shown on Fig. 5. The program languages C, Pearl and Java are used for the different functions development.

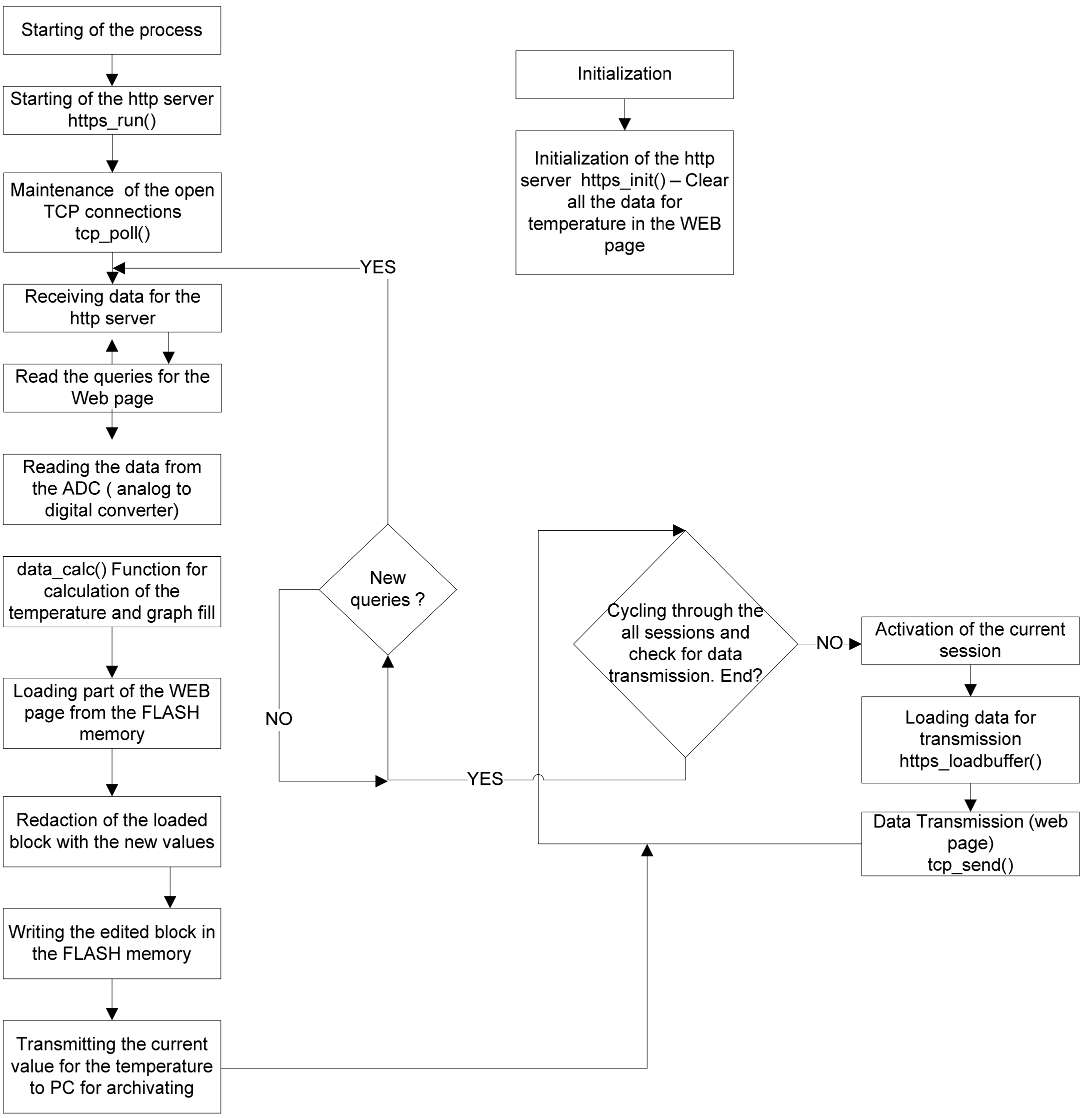

Fig. 5. General algorithm

First, when the controller is started, the WEB server is been initialized. After converting the values from the ADC the WEB page template is loaded from the FLASH memory of the controller. A screenshot from the WEB page open in the browser is shown on Fig. 6.

The HTTP page is static and only particular fields are changing in it. When the page is loaded for the first time (during initialization) all the temperature values are set to "- "and the chart is empty. The format of the data is static in the form of 00.0. The page code automatically sends queries to the controller for refresh the content of the page every 5 seconds.

This triggers the following algorithm – the main function is in a cycle which checks for queries sent over the Internet. When a query is acquired the content of the registers of the ADC is scanned. The ADC stores its data in binary format in 2 registers when it is used in 10 bit mode.

The values are converted first to decimal numbers and then to strings. The later conversion is needed because each position of the numbers on the graph has to be modified separately.

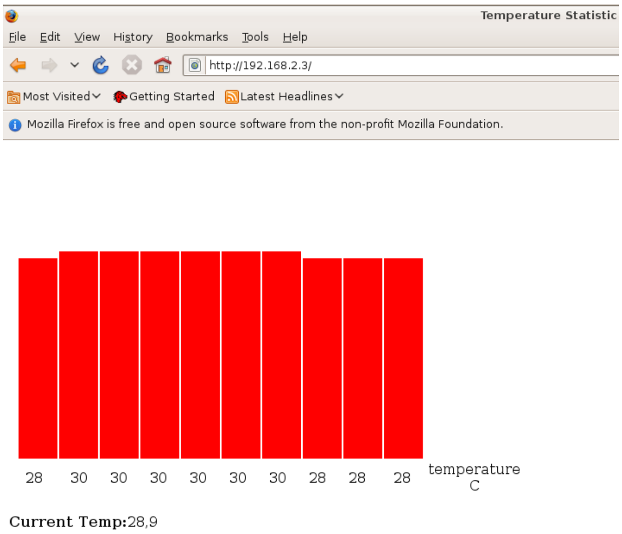

Fig. 6. WEB page template

The code checks which value on the bar chart will be modified. If the chart is not full the current values are written under the graph and visualized until the last position is filled. Then when all the bars are full, when a new value has to be written all other shift left (the oldest is deleted) and the new one is placed on the empty place. In this way a moving affect on the graph is accomplished. The page template then is loaded to the RAM memory, modified and written to the FLASH again. Then the page is read from the FLASH and loaded to the buffer to be sent.

Storing the values from the measurement is available over UDP protocol and a personal computer. A screen shot of the program is shown on the Figure 7. The program includes one endless cycle. When the program is started it endlessly tries to establish a socket connection on port 5001. This is the Internet port where the microcontroller program is set to send data over UDP protocol on each page refreshment. After the socket connection is established, the data is loaded and placed in variables. Before the cycle start a text file is open for append. Before writing the data to file, the local time and date from the computer are stored in a variable. After that the date and time are written with the appropriate formatting, at the end of the file.

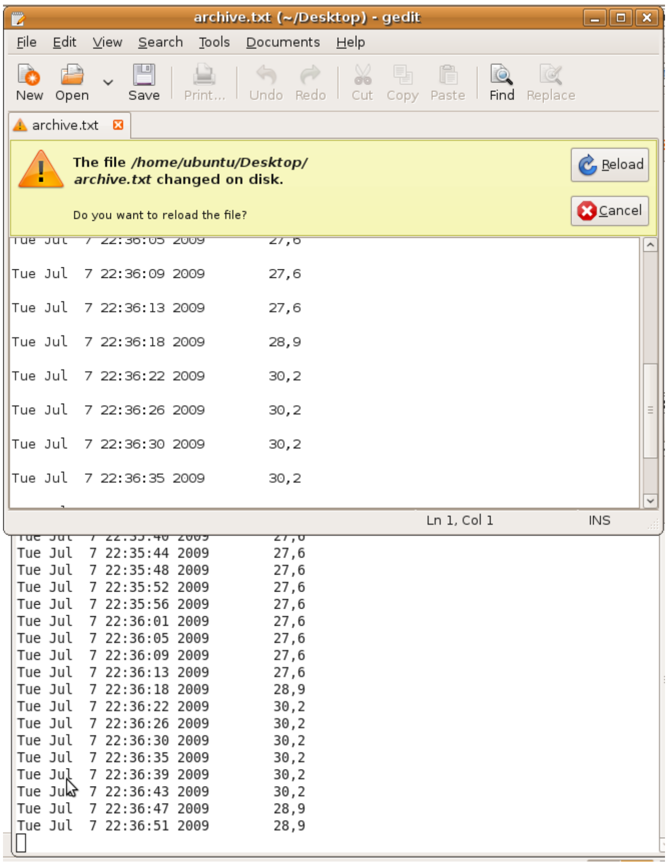

Fig. 7. Data storage program

The basic difference between TCP/IP protocol used for the web page and the UDP protocol is that the UDP is unidirectional and does not require receiving acknowledgments. This makes it faster, but not as reliable since the receiver cannot request certain package to be sent again in case of an error. A short Perl script is used to process the data sent over UDP.

III. Conclusion and Future Progress

The goals achieved with the proposed system are:

- integration of the transducer (the temperature sensor) into application in a network environment;
- real-time monitoring of the results;
- data processing on a PC, for storage and visualization of the data storage.

These goals concur with some of the IEEE 1451 standards.

The development of the system is not complete and it can be expanded to incorporate other sensors, switches and actuators allowing the system to be used in various applications like air conditioning controller, wind turbine control and others. Using it for measuring temperature is just a simple example of the recourses it possesses.

Using a WEB browser is also not necessary. A TCP/IP or UDP socket connection can be easily established with some easy to use and free programming languages. This removes some of the limitation of the web page that has resides on controller. One of the major limitations is the size of the page (a simple page as the one shown above takes 3 blocks of memory 512 bytes each).

External memory can be added to the microcontroller, preventing data loss due to Internet connection failure. Since the program code of the controller does not add data and time to the bar graph (this is made by the activation program) the timers of the controller should be use that case.

REFERENCES

[1] Herb Schildt's C Programming 2001, SoftPress Ltd.

[2] Martin C. Brown Perl: The Complete Reference Second Edition.

[3] Eugene Y. Song and Kang Lee - Understanding IEEE 1451 Networked Smart Transducer Interface Standard, Instrumentation & Measurement Magazine April 2008.

[4] National Semiconductor, LM35 Precision Centigrade Temperature Sensors, November 2000.

[5] Kakanakov N., G. Spasov, Adaptation of Web service architecture in distributed embedded systems, Proceedings of the International Conference on Computer Systems and Technologies - *CompSysTech' 2005*, 16-17 June 2005, Varna, Bulgaria, pp. IIIB.10-1– IIIB.10-6, ISBN 954-9641-38-4.

[6] http://www.freescale.com