

Learning Remote Temperature Measurement in Instrumentation Laboratory

L.Breniuc, C.G. Haba, A.Salceanu

"Gh.Asachi" Technical University of Iasi, Bd.D.Mangeron Nr.53, 700050 Iasi, Romania
E-mail: lbreniuc@ee.tuiasi.ro, cghaba@ee.tuiasi.ro

Abstract – In our paper we present a laboratory temperature measuring system that can be remotely accessed by Internet. The system is based on the DS18B20 and DS1822 digital temperature sensors controlled with the aid of a microcontroller board PCB80c552. The system is used to measure and monitor the temperature in different places inside a computer. Using a web browser we can remotely configure the operating mode of each temperature sensor as well as we can read measured temperature in each measuring point. Using this system student can learn: use of integrated temperature sensors, software implementation of a 1-Wire interface, distributed measurement systems based on 1-Wire interface, microcontroller based instrumentation, web based monitoring and control of measurement systems.

I. Introduction

With the recent technological development, contemporary measurement science evolves very rapidly. Students must be helped to develop the theoretical background and practical skills in order to cope with advanced methods and theories.

Modern computers combined with recent development of Internet give the possibility to build systems that can be accessed from remote location providing interactivity and animation, offering a flexible, faster and friendlier learning [1,2]. The increased use of computers into laboratory, brought important benefits among which we point-out: making the computer an integral part of the educational process (replacement of traditional laboratories with virtual or remote laboratories), automating measuring tasks, out-of-class assignments, etc.

The present paper presents a temperature measuring system that can be accessed by Internet.

II. Equipment and software

The temperature measuring system consists of following equipment:

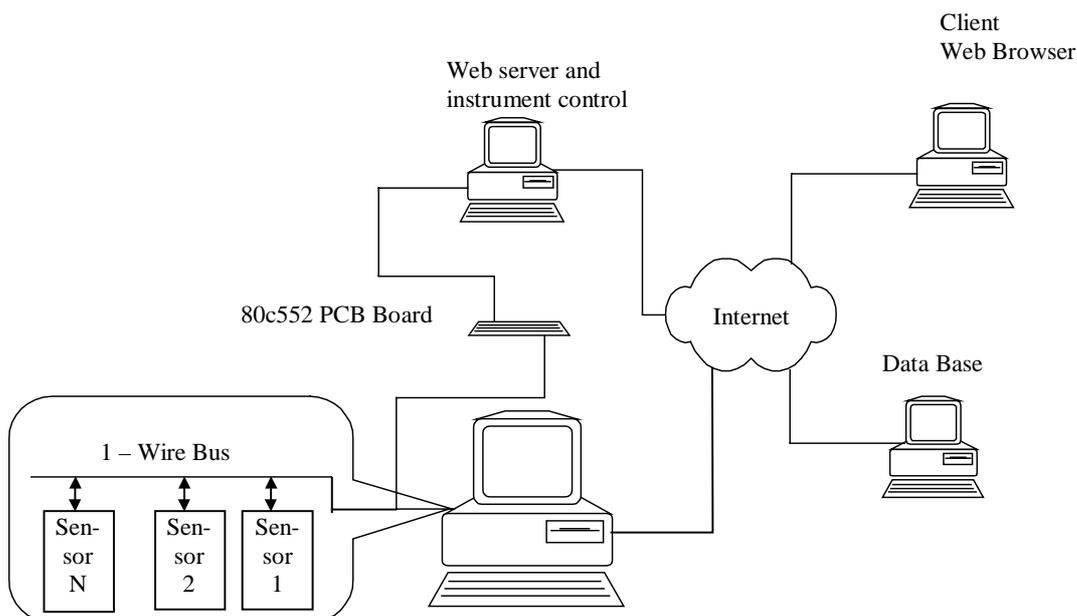


Figure 1. The architecture of the web-based temperature measurement system.

- PCB80c552 board - a microcontroller development board based on the 80c552 microcontroller.
- temperature sensors: four digital temperature sensors (three DS18B20 sensors and one DS1822 sensor).
- PC PentiumIV: the PC whose temperature is monitored in four different places,
- PC PentiumIV - web based server configured with RedHat Linux version 8.0 and running the Apache v.1.3.22 HTTP server.

The microcontroller program is written in assembler using Franklin software development tools.

The remote configuration and reading of the temperature sensors is done using CGI scripts written in C language, [3].

We can operate remotely the monitored PC using AT&T VNC software. VNC is a remote display system that allows you to view the desktop of a remote machine running, anywhere on the Internet, [4].

III. Implementation

The 1-Wire interface was initially designed for communication between devices situated at short distances one from another and as a mean for attaching an auxiliary memory using a single pin of a microprocessor port. The interface has been continuously improved and today the distance between the master and slave devices can be 750m long. A large number of devices are now compatible with the 1-Wire interface: sensors (for temperature, humidity, pressure, wind speed and direction etc), AD converters, DA converters, memories, real time clocks etc. Using this kind of devices addressable digital instruments (ADI) can be conceived. For example, a weather station is described in [5].

The system we have realized uses four temperature sensors of which three are DS18B20 sensors and one is a DS1822 sensor, [6,7]. The difference between the two sensor types is that the last one has an accuracy of $\pm 2^{\circ}\text{C}$ for temperature measurement in comparison with the first type which has an accuracy of $\pm 0.5^{\circ}\text{C}$.

The main characteristic features of these sensors are:

- 1-Wire interface compatible,
- each sensor has a unique 64 bit address. In this way on the same interface can be connected several sensors and even different 1-Wire compatible devices,
- temperature measurement resolution is programmable within the range of 9-12 bits,
- user-definable alarm settings using an EEPROM memory.

We have developed our web-based system in order to work on a network architecture like the one depicted in Fig.1. The measurement system can be operated locally or remotely. The sensors are connected to a serial bus using the 1-Wire interface for communicating with the microcontroller.

In our system we use four temperature sensors to measure and monitor temperature in four points inside the case of a PC. The three DS18B20 sensors are supplied from an external power supply and the DS1822 sensor is supplied using the parasite power mode (see Fig. 2).

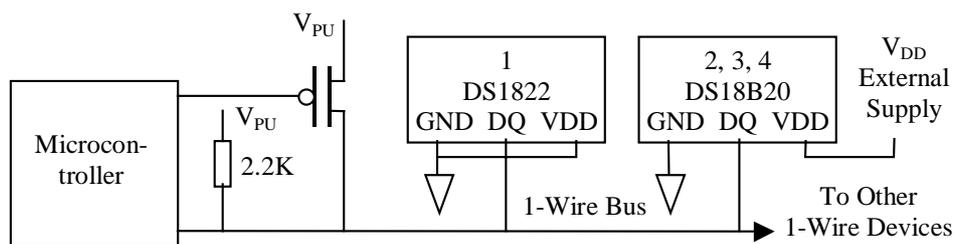


Figure 2. Sensor connection to the 1-Wire interface.

The system can be extended to use any number of sensors and can be used to monitor the temperature in other applications.

Remote monitoring and control of the sensors is achieved by setting a communication link between the microcontroller board and the web server. In the present implementation the microcontroller board is connected to the PC via a RS232 serial link.

The master device (microcontroller) communicates with the internal memory of the sensors using the DQ line. The internal sensor memory is presented in Fig. 3.

The internal sensor memory is organized in three sections:

- the ROM memory (ROM CODE) it is 8 byte in size and contains the unique address of the sensor (1 byte for the family code, 6 bytes for the serial number of the device and 1 byte

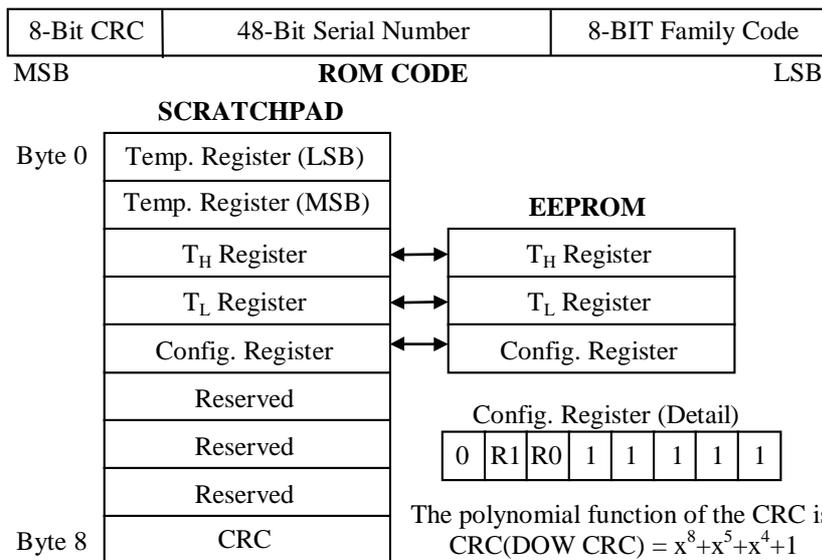


Figure 3. Internal memory of DS18B20 and DS1822 sensors.

that contains the CRC).

- the RAM memory, denoted as the SCRATCHPAD is 9 byte in size and contains 2 read-only temperature registers, 2 read-write registers that store the high (T_H) and low (T_L) alarm levels for the temperature, a read-write configuration register that sets the resolution of temperature measurement (R1R0=00 – 9bit, ..., R1R0=11 – 12 bit), 3 reserved 1-byte registers and the read-only CRC register.
- the EEPROM memory is 3 byte in size and mirrors TH, TL and configuration registers in the RAM memory.

The measurement process is controlled by the PC via the serial interface RS232, the data being transferred in ASCII code. Thus, when reading or writing a byte from/to the microcontroller, two ASCII characters will be transferred.

The communication protocol between the PC and the microcontroller is described in Figure 4.

In Fig. 4, character # represents a number between 0 to 255 (one byte) and will be represented using 2 ASCII characters.

In the following we give some examples of communication between the PC and the microcontroller board indicating the command sent by the PC and the answer received from the microcontroller.

- SEARCH (The master finds out the number and type of the slave devices connected to the 1-Wire interface).

The PC sends the command **s(0ah)**, where 0ah indicates the end of the command. Receiving this command the microcontroller runs the SEARCH algorithm which identifies the number of slave devices and their addresses. At the end of the SEARCH algorithm, the microcontroller will return to the PC the sequence **s#aabb...xx(0ah)**. For example, s0428282822(0ah) means that there are 4 slave devices (04), where three are DS18B20 devices (family code 28h) and one is a DS1822 device (family code 22h). The microcontroller board does not send the whole address of the device but only the family code (the device address is stored locally in the microcontroller memory).

- READ ALL (The microcontroller reads the SCRATCHPAD memory of all sensors connected to the 1-Wire bus detected in the previous command).

The PC sends command **r(0ah)**. Receiving this command the microcontroller executes the following steps:

- send SKIP ROM command (addressing all sensors);
- send CONVERT T command (initiate the temperature measuring process for all sensors);
- address first sensor (command MATCH ROM00), reads SCRATCHPAD memory for the first sensor (command READ SCRATCHPAD), stores the data read from the SCRATCHPAD memory of the first sensor;
- repeat addressing and read commands for all sensors;
- send to the PC the sequence **r#aabbccddeer#.....(0ah)**. For example, if the sequence is r0050054B467Fr01000560401F(0ah), this means that there are two sensors and the data read

from these sensors are: sensor 00 - temperature = 85 °C, $T_H = 75$ °C, $T_L = 70$ °C, 12-bit resolution; sensor 01- temperature = 80 °C, $T_H = 96$ °C, $T_L = 64$ °C, 9-bit resolution.

The commands can be sent manually to the microcontroller board using the serial port and a terminal emulator (for example TERA TERM PRO).

PC Command	Microcontroller Board Command Sequence	Microcontroller Board Reply
s(0ah) (SEARCH)	SEARCH ROM	s#aabb...xx(0ah) # - the number of sensors (two ASCII characters), aa...xx – two ASCII characters indicating device family
a(0ah) (ALARM SEARCH)	ALARM SEARCH	1) a##..#(0ah) # - the number of the sensor in alarm state (example: 00, 04, 80 – three sensors, 0, 4 and 8 in alarm state). If 2) a(0ah) – no sensor in alarm state
p(0ah) (READ POWER SUPPLY)	1. SKIP ROM, READ POWER SUPPLY 2. MATCH ROM00, READ POWER SUPPLY; MATCH ROM01, READ POWER SUPPLY; ... Observation: Step 2 is executed only if at least one sensor uses the parasite power mode	1) p##..#(0ah) # - the number of the sensor in parasite power mode (example: 00, 08 – sensor 0 and sensor 8 in parasite power mode) 2) p(0ah) - no sensor with parasite power mode
w#aabbcc(0ah) (INDIVIDUAL CONFIGURATION)	MATCH ROM#, WRITE SCRATCHPAD ($T_H = aa$, $T_L = bb$, Config = cc)	w#(0ah) - at the end of configuration
waabbcc(0ah) (CONFIGURE ALL)	SKIP ROM, WRITE SCRATCHPAD ($T_H = aa$, $T_L = bb$, Config = cc) if sensor configuration is the same for all sensors.	w(0ah) - at the end of configuration
c#(0ah) (INDIVIDUAL COPY SCRATCHPAD)	MATCH ROM#, COPY SCRATCHPAD	c#(0ah) – at the end of copy command
c(0ah) (COPY SCRATCHPAD ALL)	SKIP ROM, COPY SCRATCHPAD	c(0ah) - at the end of copy command
e#(0ah) (INDIVIDUAL RECALL E²)	MATCH ROM#, RECALL E ²	e#(0ah) – at the end of recall command
e(0ah) (RECALL E² ALL)	SKIP ROM, RECALL E ²	e(0ah) – at the end of recall command
r#(0ah) (INDIVIDUAL READ SCRATCHPAD)	MATCH ROM#, CONVERT T, MATCH ROM#, READ SCRATCHPAD	r#aabbccdde(0ah) aa - temperature low, bb- temperature high, cc – T_H , dd- T_L , ee - Config
r(0ah) (READ ALL)	SKIP ROM, CONVERT T, MATCH ROM00, READ SCRATCHPAD; MATCH ROM01, READ SCRATCHPAD;	r#aabbccddeer#.....(0ah)

Figure 3. Internal memory of DS18B20 and DS1822 sensors.

An HTML form was created for interfacing the temperature measurement system and the remote user connected to the Internet using a web browser.

The interface is divided in two sections:

- the measurement section (upper part of Fig.4a) - allows the user to read current temperature of each sensor separately or of a group of sensors by checking the corresponding check boxes and pressing the Submit button.
- the configuration section (lower part of Fig.4a) - allows the user to configure different parameters of the sensors, individually or for a group of sensors, by selecting available options and checking the corresponding check boxes.

Measurement

Sensor	TS1	TS2	TS3	TS4
Type	DS18B20	DS18B20	DS18B20	DS18B22
Current Value	25.25°C	!34.25°C!	!35.50°C!	24.75°C
Resolution	9-bit	9-bit	9-bit	10-bit
Power Supply	External	External	External	Parasite Power
Alert				
Read All <input type="checkbox"/>	Read TS1 <input type="checkbox"/>	Read TS2 <input type="checkbox"/>	Read TS3 <input type="checkbox"/>	Read TS4 <input type="checkbox"/>

SUBMIT

Configuration

Upper Limit (T _H)	<input type="text" value="30.00"/>	<input type="text" value="30.00"/>	<input type="text" value="30.00"/>	<input type="text" value="30.00"/>
Lower Limit (T _L)	<input type="text" value="20.00"/>	<input type="text" value="20.00"/>	<input type="text" value="20.00"/>	<input type="text" value="20.00"/>
Resolution	9-bit	9-bit	9-bit	10-bit
Configure All <input type="checkbox"/>	Configure 1 <input type="checkbox"/>	Configure 2 <input type="checkbox"/>	Configure 3 <input type="checkbox"/>	Configure 4 <input type="checkbox"/>

a)

Measurement

Sensor	TS1	TS2	TS3
Type	DS18B20	DS18B20	DS18B22
Current Value	25.50°C	!34.25°C!	24.25°C
Resolution	9-bit	9-bit	10-bit
Power Supply	External	External	Parasite Power
Alert			
Read All <input type="checkbox"/>	Read TS1 <input type="checkbox"/>	Read TS2 <input type="checkbox"/>	Read TS3 <input type="checkbox"/>

SUBMIT

b)

Figure 4. The Html form for reading and configuration of temperature sensors using a Web-browser:
 a) 4 sensors connected to the 1-Wire bus, b) 3 sensors connected to the 1-Wire bus.

At the opening of the Html form or when submitting the Read All command, a SEARCH command is send to the microcontroller in order to determine the configuration of the sensor network. If a sensor is removed from the network, the Html form is adjusted accordingly. In Fig. 4b, the Html form reflects the situation when one of the DS18B20 sensors was removed from the system.

If the alarm setting is triggered, the value of the current temperature changes colour to red and also the graphic icon for the alert changes shape and colour in order to make the alert status more visible.

The web-based interface is easy to use because it includes well-known form controls: text inputs, check box or single select lists. Experiments are carried out to determine the temperature variation related to the type of application software run on the monitored computer (calculation, graphic processing, multimedia, etc.). We can start remotely various test-bench application software by running a VNC (Virtual Network Computing) server on the monitored computer. Using the VNC viewer (the VNC corresponding client) we get access to the monitored computer desktop.

IV. Conclusions

Using our system, students can carry out experiments regarding distributed temperature measurement inside or outside the laboratory using a remote computer connected to Internet. Further work will seek to implement shared and secure access to our system.

References

- [1] B.Aktan, C.A.Bohus, L.A.Crowl, M.H. Shor, "Distance Learning Applied to Control Engineering Laboratories", *IEEE Trans Education*, vol. 39, no. 3., pp. 320-326, 1996.
- [2] C. McCormack, D. Jones, *Web-Based Education System*, John Wiley & Sons Inc., 1977.
- [3] Thomas Boutell, cgic: an ANSI C library for CGI Programming., Boutell Inc., <http://www.boutell.com/cgi>.
- [4] *Virtual Network Computing*, AT&T Laboratories, Cambridge, 1999, <http://www.uk.research.att.com/vnc/>
- [5] D. Awtrey, "The 1-Wire Weather Station", *Sensors*: pp. 34-40, June 1998.
- [6] DS18B20 Programmable Resolution 1-Wire Digital Thermometer, Data Sheet, <http://www.maxim-ic.com>
- [7] DS1822 Econo 1-Wire Digital Thermometer, Data Sheet, <http://www.maxim-ic.com>