

# Learning System for Evaluation of Temperature and Humidity Sensors

Liviu Breniuc, Cristian-Győző Haba and Valeriu David

*Faculty of Electrical Engineering, Bd. D.Mangeron Nr.53, 700050, Iasi, 0040-232-278683,  
{lbreniuc,cghaba,valdavid}@ee.tuiasi.ro*

**Abstract-** In our paper we present a laboratory measuring system that students can use to learn different topics related to temperature and humidity sensors. The system consists of three different temperature sensors and a temperature/humidity sensor, all controlled with the aid of a development board based of the PIC18F452 microcontroller. Using this system student can learn: use of integrated temperature sensors, software implementation of different communication interfaces (I<sup>2</sup>C, 1-wire, CMOSens), distributed measurement systems based on these interfaces, microcontroller based instrumentation, data logging and control of measurement systems.

## I. Introduction

Recent technological development in microelectronics and computer science and their use in the measurement applications imposed the merge of traditional sensors with digital interfaces.

The interfaces that are used the most are serial because of the following advantages:

- galvanic isolation,
- reduced number of wires,
- configurable transmission rate,
- reduced complexity of the measurement system.

Serial interface used in sensors can be standardized (I<sup>2</sup>C, SPI, 1-wire) or can be proprietary (CMOSens, PWM based).

Students must be helped to develop the theoretical background and practical skills in order to cope with continuously increasing number of measurement systems based on digital sensors.

The present paper presents a temperature and humidity measuring system that can help students learn use of integrated temperature sensors, software implementation of different communication interfaces (I<sup>2</sup>C, 1-wire, CMOSens), distributed measurement systems based on these interfaces, microcontroller based instrumentation, data logging and control of measurement systems.

## II. Measurement system

The measuring system consists of following equipment:

- PICDEM 2 PLUS board - a microcontroller development board based on the PIC18F452 microcontroller [1].
- temperature and humidity sensors:
  - 10-bit digital temperature sensors (AD7416) [2],
  - 8 - bit bit digital temperature sensors (TC74) [3],
  - 9-12 bit digital temperature sensor (DS18B20) [4],
  - 12-14 bit temperature and humidity sensor (SHT11) [5];
- PC Pentium – used to collect data from sensors and display on the screen in different formats.
- temperature and humidity data logger (RH520) used to compare the measurements.

The microcontroller programs are written in assembler or C languages using Microchip MPLAB IDE software development tools [6]. The Proteus software [7] was also used to simulate part of the measurement system.

Communication of measured data is done using different interfaces using standard and proprietary communication protocols: I<sup>2</sup>C for AD7416 and TC74, 1-wire for DS18B20 and CMOSens for SHT11.

Using this system students will learn how to interface sensors to microcontroller systems, communication protocols (two standard and one proprietary). The interfacing can be implemented using dedicated peripheral in the microcontroller (I<sup>2</sup>C) or using only common I/O ports and software routines.

### III. Measuring set-up

All sensors we have chosen have different degree of intelligence. This implies that the sensors can be configured at the initialization of the system and also during the measurement protocol. In order to obtain accurate measured values special conditions must be taken into considerations.

Using the system student can learn:

- how to initialize sensors,
- how to calibrate sensors,
- how to adopt a certain resolution in concordance with the measurement application,
- how to configure different features available in each sensor,
- how different parameters influence the measurement process and what action should be taken in order to reduce the negative influences:
  - distance between sensor and data collecting system,
  - electromagnetic compatibility.

Each sensor type has an internal structure different from the others. Therefore students can learn:

- internal registers;
- available commands for reading and writing internal registers;
- sequence of commands for configuring different sensor features;
- using existing low-power features;
- using calibration features.

### IV. Sensor network

Three of the used sensors use interfaces (I<sup>2</sup>C and 1-wire) that allow the creation of a sensor networks.

The learning system allows students to learn:

- how sensor can be connected into a network having different topologies
- limits regarding connection of sensors in a network (distance, electromagnetic compatibility, power supply, communication speed etc).
- how sensors can be integrated in a larger system with other sensors or components having same interfaces and using same protocols (example a network of sensors and memory for storing the measured data)
- the important parameters that influence the performance of a network of sensor

### V. Controlling sensors

Configuration and measurement of temperature and humidity is performed by a microsystem based on the PIC18F452 microcontroller. Using the system students can perform only measurement activities using existing programs to control the sensors but they can also develop their own programs.

They learn how to write programs to control the sensors either using assembler or C language.

In their programs they can use existing interfaces or can implement the protocols for a certain interface in software.

### VI. Simulation

Modern verification methods for microcontroller based applications include simulation. In our case we were able to use the Proteus software to simulate part of the measuring system.

Proteus software consists of several modules which allow the input of electric diagram, SPICE simulation of analogue and digital devices and the design of the printed circuit board (PCB). One special feature is the ability to simulate running programs for different microcontroller families.

The electric diagram for simulation inputted into the Proteus software is depicted in Figure 1 and it follows the structure of the development board PICDEM 2 PLUS. The system is built around PIC18F452 microcontroller. The diagram also includes the temperature sensors TC74 (I<sup>2</sup>C compatible) and DS18B20 (1-wire compatible) connected to the ports of the microcontroller. We could not simulate AD7416 and SHT11 because we didn't have the software models to be used with the Proteus software. The following components are also present into the diagram:

- digital potentiometer MCP41010 (SPI compatible). This device is useful for the study of SPI interface.
- green LED D5 for power on indication

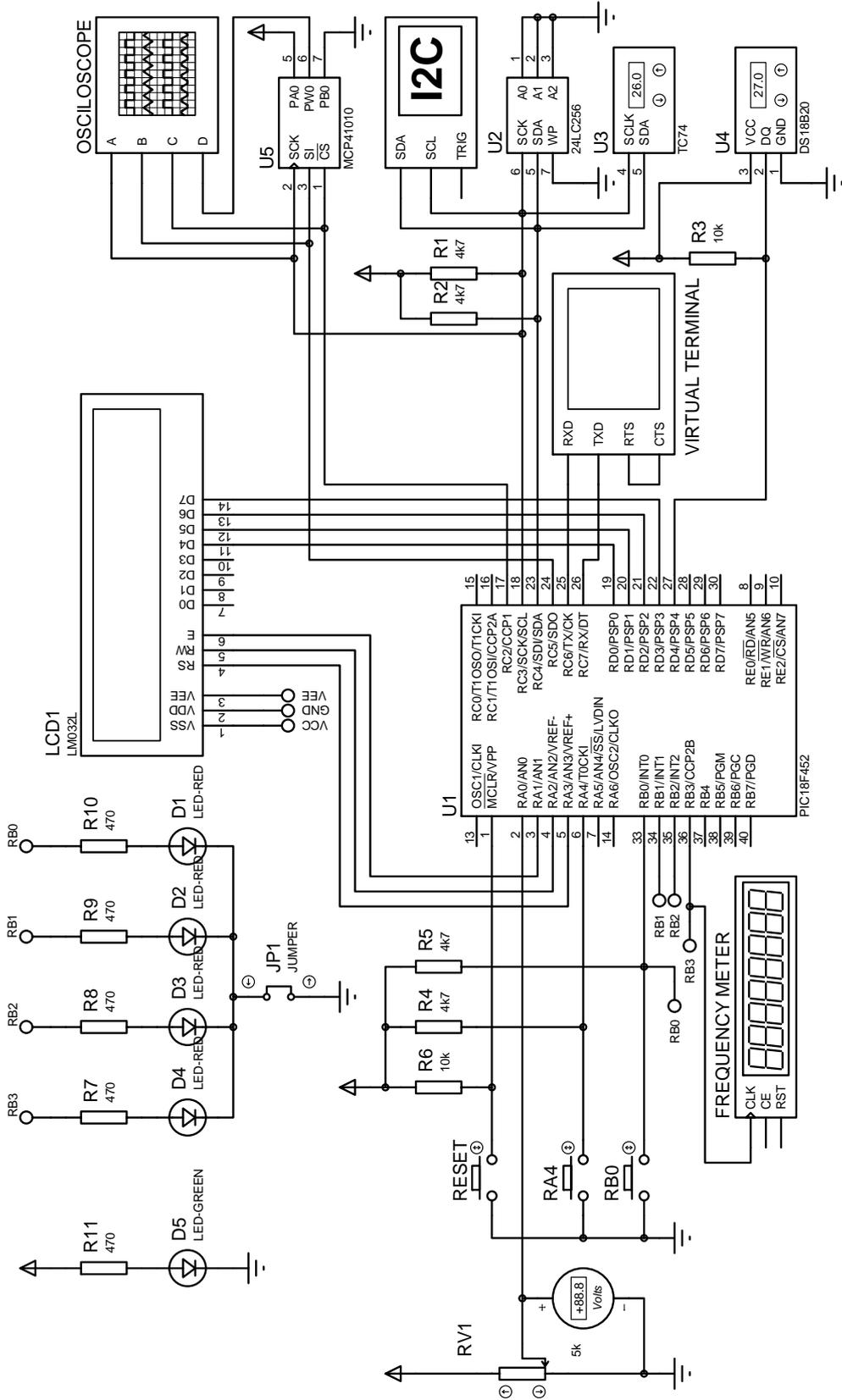


Figure 1. The electric diagram for simulation red LEDs connected to the RB0-RB3 pins of the microcontroller. They can be used to observe logic levels at these pins and can be suppressed with JP1 jumper.

- RESET, RA4 and RB0 animated buttons. These buttons can be pushed to simulate inputs to the corresponding microcontroller inputs
- RV1 animated potentiometer. The value of the potentiometer can be modified during simulation and can be used to test the microcontroller analog to digital converter (ADC).
- 2x16 characters LCD display.

In order to be able to verify system operation and microcontroller running user application we can also connect virtual measurement devices.

- DC Voltmeter connected to the cursor of the RV1 potentiometer
- Frequency meter connected the RB3/CCP2 output pin
- Oscilloscope connected to the pins of the MCP41010 digital potentiometer
- Virtual Terminal connected to the Tx and Rx of microcontroller UART port pins
- I2C Debugger connected to the SCL and SDA pins of the I<sup>2</sup>C port of the microcontroller.

The verification of an application implies the following steps: design of the electric diagram of the application (example Figure 1), writing the microcontroller program (in assembler or in C), assemble or compile program for obtaining executable files (.hex or .cof), chose working frequency, load (attach to simulator) the .hex or .cof executable files, simulate application.

We will present the results for the simulation of temperature sensors TC74 and DS18B20.

The sensor TC74 produced by Microchip has the following characteristics: measurement range: -40 - +125 °C; resolution: ± 1 °C (8 bit), precision: ± 2 °C, number of conversions: 4-8/s, compatibility with I<sup>2</sup>C interface, 1001101 – slave address. Working with the sensor implies using two registers: RTR – temperature register and CONFIG – control register. Each register has its own address, 00 for RTR and 01 for CONFIG. Temperature register contains the measured temperature on 8 bit in 2 complement format. The CONFIG register contains two active bits, bit 7 – SHDN and bit 6 – DATA\_RDY. DATA\_RDY=1 signify that the measured temperature is available in the RTR register. When SHDN bit is set the sensor goes in low power Standby mode.

Figure 2 depicts the content of the CONFIG and TEMP (RTR) registers during system simulation and a simulated temperature of 26 °C. In Figure 3 we present the I<sup>2</sup>C interface operation sequences generated by the I<sup>2</sup>C debugger.

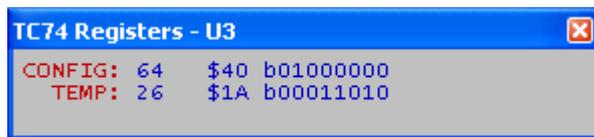


Figure 2. Content of CONFIG and TEMP (RTR) registers



Figure 3. Measured temperature display

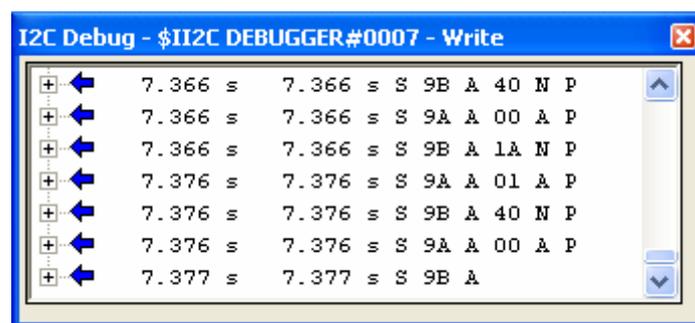


Figure 4. I<sup>2</sup>C interface operation sequences.

Sensor DS18B20 produced by Dallas has the following characteristics: measurement range: -55 - +125 °C, programmable resolution: (± 0.5 °C - 9 bits, ± 0.25 °C - 10 bits, (± 0.125 °C - 11 bits, (± 0.0625 °C - 12 bits), conversion time depends on resolution: 93.75 ms (9 bit) -750 ms (12 bit), compatibility with 1 – wire interface. This sensor is interesting because communication with the microcontroller is done using only one wire. Sensor can also take his power from this line. Using the 1 –wire interface we can connect several sensors or several devices with compatible interface. The interface supports different topologies in all topologies each device having a uniqe address. The system supports the implementation of different topologies so students can test different applications and become familiar with this interface type.

Figure 5 depicts operation sequences for the DS18B20 sensor.

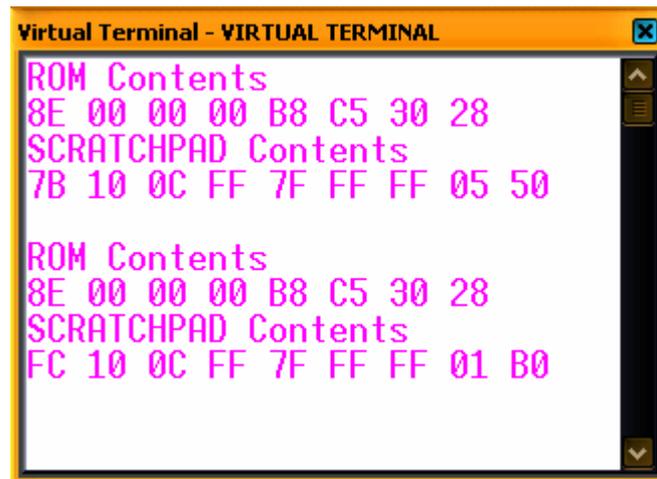


Figure 5. DS18B20 sensor operation sequences in Virtual Terminal.

Every device compatible with the 1-wire interface contains an 8 byte ROM memory. The ROM contents of the DS18B20 sensor was read and displayed in the Virtual Terminal. The meanings of the 8 bytes are the following: 28 – DS18B20sensor family code, 00 00 00 B8 C5 30 – 6 byte sensor serial number, 8E – CRC checksum of the previous bytes.

DS18B20 sensors contain also a 9 bytes RAM memory (SCRATCHPAD). This memory was read twice and content displayed in Virtual Terminal. The meanings of bytes from the Scratchpad memory are: 05 50 – 85 °C temperature (default), 7F – content of the Configuration Register (here we set measurement resolution, 7F – 12 bit resolution), 7B – is the CRC checksum computed using all bytes.

The meaning of the other bytes can be found in the sensor datasheet [4]. The second Scratchpad memory dump contains data after a conversion command. In this case the measured data is 01 B0 which corresponds to 27 °C and normally modifies the CRC (FC). The content of the Scratchpad memory can be compared to that of the content displayed in the sensor model window, Figure 6.

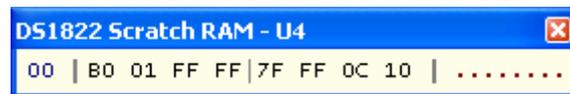


Figure 6. SCRATCHPAD memory simulation model window

## VII. Hardware tests

An important part in developing engineering skills in our students is testing developed application in real hardware. We have implemented the measurement system using a Microchip PICDEM 2 PLUS development board. The application code is loaded in the microcontroller memory using the ICD2 debugger/programmer. Students also learn how to load executable files in the memory of the microcontroller using a bootloader or a monitor. These solutions are described in [1,8].

Applications simulated with the Proteus simulation software are easily tested in the hardware. Other simulation methods must be used to pre test and simulate application that use sensor that do not have a simulation model in Proteus software. In the case of AD7416 and SHT11 sensors the applications are debugged using one of the following tools: MPLAB IDE SIM simulator, the ICD2 debugger or the monitor described in [8]. We present the results obtained when testing the application for measuring the temperature and humidity using SHT11 sensor. The results are displayed on the PC display using the TERA TERM terminal emulator, Figure 7.

Results are displayed in a 5 line format. First line – T(hexa) – contains temperature reading sequence: 03 – command word to SHT11 for temperature measurement, 1B B7 – sensor answer after temperature measurement, 69 – the checksum generated by the sensor and computed with the first three bytes. Second line – T(°C) – the computed temperature (30.9) based on sensor answer. Line three – RH(hexa) – contains humidity reading sequence: 05 – command word to SHT11 for humidity measurement, 04 C6 – sensor answer after humidity measurement, B0 – the checksum generated by the sensor and

computed with the first three bytes. Lines for and five – RHlin(%) and RH(%) – display the values for the humidity computed with the value from the sensor answer. The last humidity value is obtained from the first one applying corrections necessary for measurements that are made when temperature differs from 25 °C. Correction relation is provided by the manufacturer and is given in [5]. Students have the possibility to learn about sensor precision and sensibility by comparing the measured results using the studied sensor with the data provided by a professional temperature and humidity data logger (RH520).

```

Tera Term - COM1 VT
File Edit Setup Control Window Help
T(hexa) 03 1B B7 69
T(°C) = 30,9
RH(hexa) 05 04 C6 B0
RHlin(%) = 41,3
RH(%) = 41,9

T(hexa) 03 1B BB D7
T(°C) = 30,9
RH(hexa) 05 04 C9 C4
RHlin(%) = 41,4
RH(%) = 42,0

```

Figure 7. Temperature and humidity measured data read from SHT11 sensor.

## VIII. Conclusions

In this paper we have presented a laboratory system which students can use to learn intelligent temperature sensors, microcontroller based measurement systems that control these sensors and communication protocols necessary to control and transfer data to and from these sensors. The presented system supports several laboratory works with different degrees of complexity and which develops practical skills at different levels, important for establishing a good engineering education for the participant students.

The system is also available in a simulation model based on the Proteus software which can be used in a preliminary stage before starting to work with the hardware platform either to demonstrate the simulation step in a typical design flow or to reduce system damages in early stages when students do not possess enough practical skills.

## Acknowledgements

This paper has been supported by the Romanian Ministry of Education and Research under the Project CEEX-M1-C2-312 (MATNANTECH 46/26.07.2006) and the Project CEEX-M3-C3-12527 (CNMP 202/2006) in the frame of the CEEX Program.

## References

- [1] L. Breniuc, C. G. Haba, *Design with PICmicro Microcontrollers*, Politehniun, Iasi, 2005, ISBN 973-621-130-4
- [2] \*\*\*, *10-Bit Digital Temperature Sensor (AD7416) and Single/Four-Channel ADC (AD7417/7418)*, Analog Device Inc, 2000.
- [3] \*\*\*, *TC74 Tiny Serial Digital Thermal Sensor*, DS21462C, Microchip Technology Inc, 2002
- [4] \*\*\*, *DS18B20 Programmable Resolution 1-Wire Digital Thermometer*, Data Sheet, <http://www.maxim-ic.com>
- [5] \*\*\*, *SHT1x-SHT7x Humidity and Temperature Sensor*, Sensirion A.G., 2006.
- [6] \*\*\*, *MPLAB IDE User's Guide*, DS51519B, Microchip Technology Inc., 2006.
- [7] \*\*\*, *Proteus Design Suite User's Guide*, Labcenter Electronics Ltd., 2007.
- [8] J. B. Peatman, *Embedded Design with the PIC18F452 Microcontroller*, Prentice Hall, 2003, ISBN 0-13-046243-6