BULETINUL INSTITUTULUI POLITEHNIC DIN IAȘI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iași
Tomul LX (LXIV), Fasc. 1, 2014
Secția
ELECTROTEHNICĂ. ENERGETICĂ. ELECTRONICĂ

MONITORING THE SHT75 RELATIVE HUMIDITY SENSOR USING AN ATMEL FAMILY MICROCONTROLLER

BY

PETRUȚ DUMA*

"Gheorghe Asachi" Technical University of Iași
Faculty of Electronics, Telecommunications and Information Technology

Received: February 12, 2014
Accepted for publication: March 25, 2014

Abstract. The paper describes the hardware interface required for measuring the relative humidity and temperature, using the digital humidity sensor SHT75 and a development system equipped with an ATMEL family microcontroller. The command program measures and displays the relative humidity and the temperature, compensates the measured values, converts the data read from the sensor into numerical values with a physical significance, sends the acquired data to a personal computer and performs a wide range of processing on the obtained data.

Key words: digital relative humidity sensor; ATMEL microcontroller; hardware interface; application system; command program.

1. Introduction

The relative humidity sensor, such as other transducers, *e.g.* those for pressure, temperature, etc., converts the variations of the measured physical property into variations of an electrical property. These properties are amplified and processed using specific analogue integrated circuits. Finally, using converters and digital processing for scaling and compensation, the numeric value of the measured physical property is obtained.

Another sensor category allows to measure the required physical

* *e-mail*: pduma@etti.tuiasi.ro

property and to process the analogue signal on the same chip. Implementing these sensors is limited to using the transducer and the analogue processing circuits that output a current or a voltage proportional to the measured property.

Nowadays, digital sensors are used, that include on the same chip the transducer, the amplifying and analogue processing circuit, the analogue–digital converter, compensation circuits, command and control circuits, a serial communications interface and other specific circuits, if required.

The basic structure of a relative humidity monitoring system, used in fact to monitor also other physical properties such as pressure, temperature and so on, controlled by an application system equipped with an ATMEL family microcontroller, is represented in Fig. 1. The notes have the following meaning: RHS – relative humidity sensor, RHSI – relative humidity sensor interface, PS – pressure sensor, PSI – pressure sensor interface, TS – temperature sensor, TSI – temperature sensor interface, µC_AS – microcontroller equipped application system, RTC – real-time clock, DDC – data display console (LCD), SFM – serial FLASH memory, SI_RS232 – serial interface RS232, PC – personal computer.

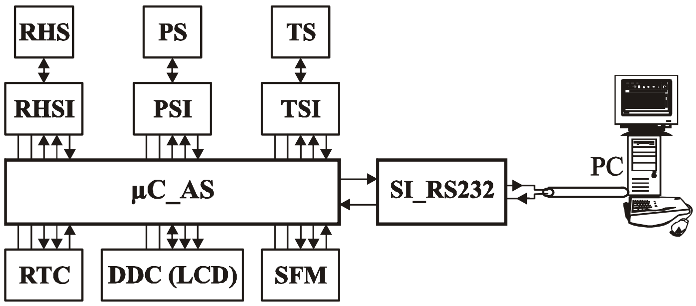

Fig. 1.

The serial FLASH memory stores the values of the measured properties at certain scheduled time intervals, their extreme values and the value of the real-time clock. The data display console is optional and continually displays on an LCD the real-time clock and the measured values. The system uses serial asynchronous communication to connect to a personal computer in order to transmit and receive data and commands, to display the measured values in various formats, to download the FLASH memory contents onto the computer, etc. This paper covers the relative humidity sensor monitoring. Following articles will present the other functions required for setting up the system.

2. The SHT75 Digital Relative Humidity Sensor

The relative humidity sensor SHT75, made by the Sensirion company, measures the relative humidity and the temperature from the environment. The main characteristics are: the relative humidity has a 12-bits/8-bits resolution, on a normal functioning range of 0% up to 95% (the extremities being 0% and 100%), with a typical accuracy of ±1.8% (up to ±3.5% at the extremities) and a repeatability of ±0.1%; the temperature is measured with a 14-bits/12-bits resolution, on a normal functioning range between –20°C and +100°C (–40°C and +123.8°C being the limits), with a typical accuracy of ±0.3°C (up to ±1.7°C at the extremities) and a repeatability of ±0.1°C; it enables the calculation of the dew point; the device communicates through a serial interface which transfers data at a speed of 100 kbps; it has an ultra low power mode (sleep mode); the digital sensor does not need external components, it does not occupy a large amount of space, it is completely calibrated, it is extremely reliable, with a long-term stability to its advantage and a resilience to external interferences.

The internal block diagram of the SHT75 digital relative humidity sensor is shown in Fig. 2; the utilized notations have the following meaning: RHS – relative humidity sensor; TS – temperature sensor; MA_1, MA_2 – measure amplifiers; AM – analogical multiplexer; ADC – analog-digital converter; CM – calibration memory; OR – output register; SR – status register; D2WSI – digital 2-wire serial interface; CRCG – cyclical redundancy check generator; CCL – command and control logic; SB – supply block that includes ultra low power mode (ULPM).

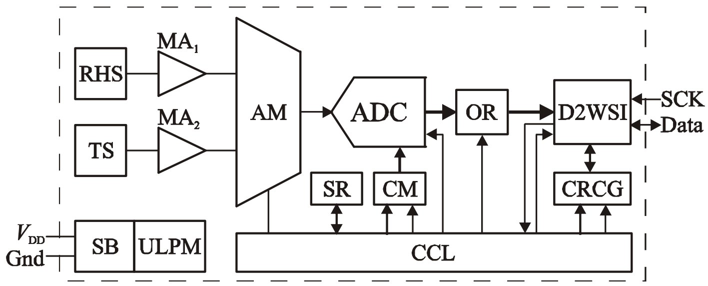

Fig. 2.

The device includes a capacitive polymer sensing element for relative humidity and a bandgap temperature sensor. The useful signals collected from the sensors are amplified, multiplexed, converted from analog to digital, the result being loaded into the output registry. The master microcontroller manages the slave sensor through a serial interface (which is not I^2C) in order to send a "Start Transmission" (ST) sequence, followed by a command for measuring the relative humidity or the temperature, then for reading of the measured values. After the execution of the compensation and processing algorithm, the significant physical values for the measured sizes are obtained.

Each device is calibrated individually in a chamber with a precisely controlled humidity level. The calibrating coefficients are programmed in an internal memory (OTP) and used during the measurements, in order to calibrate the sensors' signals.

The commands accepted by the SHT75 sensor are shown in Table 1. The most important two commands perform the measurements of the relative humidity (RH) and of the temperature (T), while the others do the writing/reading of the status register (SR) and the software reset of the sensor.

Table 1

Command Code	Parameters	Description
00011B=03H	T_{ADC}_MSB, T_{ADC}_LSB, CRC8 (read)	Temperature Measure
00101B=05H	RH_{ADC}_MSB, RH_{ADC}_LSB, CRC8 (read)	Relative Humidity Measure
00110B=06H	SR (write)	Write Status Register
00111B=07H	SR, CRC8 (read)	Read Status Register
11110B=1EH	–	Soft Reset
Other values	–	Preserved

The status register has eight bits, of which only four indicators are used: SR_0 – resolution converter ($SR_0 = 0$ means 12-bits resolution converter for RH and 14-bits for T or $SR_0 = 1$ means 8-bits resolution converter for RH and 12-bits for T); SR_1 – no reload the calibrating coefficients from OTP ($SR_1 = 0$ reload or $SR_1 = 1$ no reload); SR_2 – stops the heating element ($SR_2 = 0$ heating off or $SR_2 = 1$ heating on); SR_6 – low power supply voltage ($SR_6 = 0$ for $V_{DD} >$ > 2.47 V or $SR_6 = 1$ for $V_{DD} < 2.47$ V).

After receiving the measurement command (RH or T), the selected measure is converted, then the microcontroller reads the most significant byte (MSB) and the least significant byte (LSB) that contain the measurement result. The bytes read have only unsigned integer values, with the following format:

$$RH_{ADC}\_MSB = 0000 I_{11} I_{10} I_9 I_8 \text{ for 12 bits;}$$
$$(RH_{ADC}\_MSB = 00000000 \text{ for 8 bits);}$$
$$RH_{ADC}\_LSB = I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0;$$
$$T_{ADC}\_MSB = 00 I_{13} I_{12} I_{11} I_{10} I_9 I_8 \text{ for 14 bits;}$$
$$(T_{ADC}\_MSB = 0000 I_{11} I_{10} I_9 I_8 \text{ for 12 bits);}$$
$$T_{ADC}\_LSB = I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0,$$

where I_k is the rank k bit of the respective measure.

The numerical values of the measured relative humidity and temperature (for the maximal resolution) are:

$$RH_{ADC} = I_{11} I_{10} I_9 I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0;$$
$$T_{ADC} = I_{13} I_{12} I_{11} I_{10} I_9 I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0.$$

The low resolution of the converter is used in the speed applications when simpler calculations are made or for an extremely low power consumption.

The compensation of nonlinear response of the relative humidity sensor required in order to obtain the best precision, but also to transform the numerical value read by the sensor into a meaningful physical value, is made using the expression

$$\mathrm{RH}_{\mathrm{Lin}} = c_1 + c_2 \cdot \mathrm{RH}_{\mathrm{ADC}} + c_3\, \mathrm{RH}^2_{\mathrm{ADC}} . \quad (1)$$

The Table 2 contains the different decimal and hexadecimal values of the c_1, c_2 and c_3 constants for the V3 version of the SHT75 sensor, which depend on the resolution of the 12-bits/8-bits converter.

Table 2

$\mathrm{RH}_{\mathrm{ADC}}$	c_1	c_2	c_3	Numerical Values
12 bits	–4	0.0405	-2.8×10^{-6}	Decimal
	–4H	$\mathrm{A5E3} \times 10^{-5}\mathrm{H}$	$-\mathrm{2EFA} \times 10^{-8}\mathrm{H}$	Hexadecimal
8 bits	–4	0.648	-7.2×10^{-4}	Decimal
	–4H	$\mathrm{A5E3} \times 10^{-4}\mathrm{H}$	$-\mathrm{4B7F} \times 10^{-7}\mathrm{H}$	Hexadecimal

For values much more different from 25°C we have to take into consideration the temperature coefficient of the humidity sensor. The standard variation of the relative humidity is approximately 0.12% per Celsius degree at a relative humidity of 50%. This is done according to the following expression:

$$\mathrm{RH}_{\mathrm{T}} = (T - 25)(t_1 + t_2\, \mathrm{RH}_{\mathrm{ADC}}) + \mathrm{RH}_{\mathrm{Lin}} . \quad (2)$$

Table 3 contains the numerical (decimal and hexadecimal) values of the t_1 and t_2 constants, which depend on the resolution of the 12-bits/8-bits converter.

Table 3

$\mathrm{RH}_{\mathrm{ADC}}$	t_1	t_2	Numerical Values
12 bits	0.01	0.00008	Decimal
	$\mathrm{28F6} \times 10^{-5}\mathrm{H}$	$\mathrm{53E3} \times 10^{-7}\mathrm{H}$	Hexadecimal
8 bits	0.01	0.00128	Decimal
	$\mathrm{28F6} \times 10^{-5}\mathrm{H}$	$\mathrm{53E3} \times 10^{-6}\mathrm{H}$	Hexadecimal

The bandgap PTAT (Proportional To Absolute Temperature) temperature sensor has a good linearity by design. In order to transform the numerical value read from the sensor into a value in Celsius degrees, the following expression is used:

$$T_{^\circ\mathrm{C}} = d_1 + d_2\, T_{\mathrm{ADC}} . \quad (3)$$

The decimal and hexadecimal values of the d_1 constant that depend on the supply voltage of the sensor are shown in Table 4. The values of the d_2 constant, depending on the resolution of the 14-bits/12-bits converter are presented in Table 5.

Table 4

V_{DD}	d_1 for T in Celsius degrees	
	Decimal	Hexadecimal
5 V	–40	–28H
4 V	–39.75	$-27H - C000 \times 10^{-4}H$
3.5 V	–39.66	$-27H - A8F5 \times 10^{-4}H$
3.3 V	–39.64	$-27H - A3D7 \times 10^{-4}H$
3 V	–39.6	$-27H - 999A \times 10^{-4}H$
2.5 V	–39.55	$-27H - 8CCD \times 10^{-4}H$

Table 5

T_{ADC}	d_2 for T in Celsius degrees	
	Decimal	Hexadecimal
14 bits	0.01	$28F6 \times 10^{-5}H$
12 bits	0.04	$A3D7 \times 10^{-5}H$

3. Interfacing the Relative Humidity Sensor SHT75

The command and control of the SHT75 sensor are done with a development system, equipped with the AT89S8253 microcontroller. Fig. 3 shows the interface used for monitoring this system.

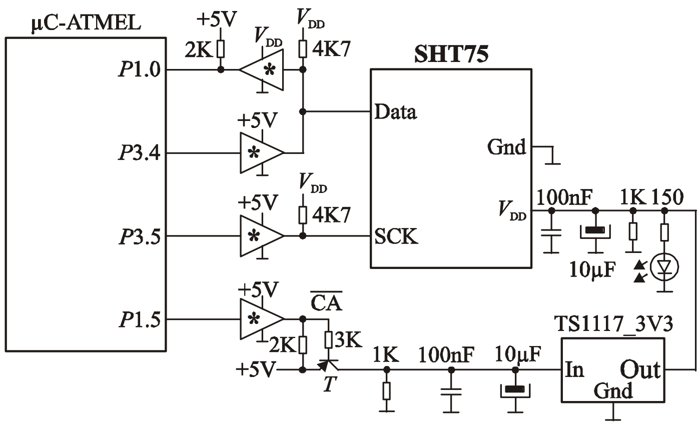

Fig. 3.

The relative humidity sensor can be powered within the DC voltage range from +2.4 V to +5.5 V. In this interface, the sensor is powered with 3.3 V from the three-point integrated positive voltage stabilizer, which only needs a few filter capacitors at its external connections.

The supply command is made through an open collector buffer and a transistor *T* (BC 556); the $\overline{\text{CA}}$ signal for this command is active on logical level “0” and is provided through software on microcontroller line *P*1.5. After powering on the sensor, the system waits for minimum 11 ms for the initialization of the internal circuits, for setting the terminals of the serial interface in high impedance state and for the status register to reset, then the device enters the sleep mode. During this time period no commands should be transmitted because they will not be taken into consideration.

The current consumed by the sensor when it runs, reaches a maximum of 0.55 mA; in sleep mode the amount consumed will not be higher than 1 μA.

The serial bi-directional data line Data and the serial clock input SCK are connected through open collector buffers to the microcontroller’s ports lines. These buffers are necessary for logical level conversion between the development system microcontroller powered at +5 V and the relative humidity sensor powered at +3.3 V. The final hardware structure of the application will be managed by an application system with a microcontroller from the ATMEL family powered at +3.3 V, and these buffers will not be necessary anymore.

4. Monitoring the Relative Humidity Sensor

The data transfers between the master microcontroller (*M*) and the slave sensor (*S*) is made through a serial interface. The clock signal (SCK), given by the microcontroller, has the role of synchronizing the communication with the sensor. Through the data line (Data), data transfers are made from the microcontroller to the sensor and inversely. The output data line from the sensor provides a logical value after the descending edge of the clock signal, while the same data line as input upload a logical value on the rising edge of the clock signal. During a transfer the Data signal should not modify during the logical “1” level of the clock signal.

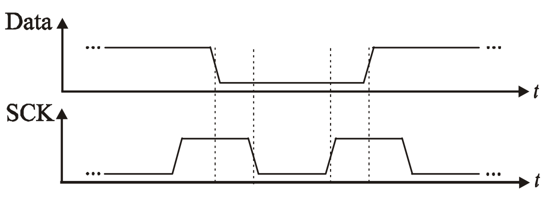

Fig. 4.

In order to initiate communication with the sensor, it is required that the microcontroller sends a “Start Transmission” sequence (TS). This consists of switching Data line from logical “1” to logical “0” while SCK is logical “1”, followed by a low pulse on the clock line, then setting the data line to logical “1” while SCK is logical “1” (Fig. 4).

Any command is initialized by the master and consists of the transmission of an eight bits command word, the first three being 000, followed by the five bits of the command code (Table 1). The command byte is transmitted bit by bit, starting with the most significant one and ending with the less significant one, during eight clock periods. The sensor confirms the command reception with an acknowledge bit (*A*) by applying a logical “0” on the Data line following the descending edge of the eighth clock period. The Data line is released and passes into logical “1” after the descending edge of the ninth clock period (Fig. 5).

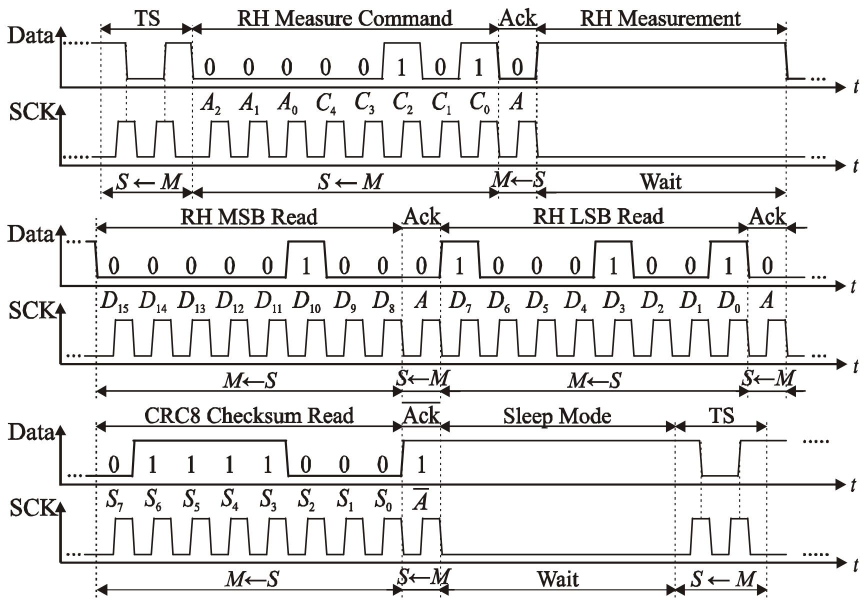

Fig. 5.

After the transmission of a measurement command, the microcontroller waits (*W*) for approximately 11 ms/55 ms/210 ms to complete the conversion, with a resolution of 8 bits/12 bits/14 bits. When the sensor finishes a measurement, it sends the logical level “0” on the data line and enters the sleep mode. The microcontroller must wait for this “data ready” signal before restarting the clock signal in order to read the data. Two data bytes are read from the sensor (MSB, LSB) bit by bit. They consist of the measured numerical value and a byte containing the checksum CRC8. The microcontroller confirms the reception of every byte of data with an acknowledgement bit (*A*) by applying a logical level “0” on the data line during the next clock period. After receiving the checksum, the microcontroller finishes the command and confirms it to the sensor with a not-acknowledge bit ($\overline{A}$) by applying a logical level “1” on the data line during the next clock period. The sensor enters the sleep mode while the microcontroller begins the data processing.

An overview with the brief evolution of data which goes through the serial interface for all commands is illustrated in Fig. 6.

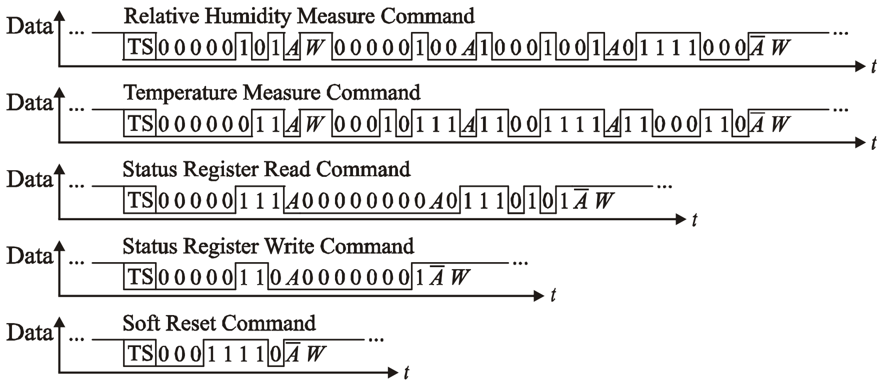

Fig. 6.

The commands that perform the relative humidity and temperature measurement are complex but similar, while other commands are simpler and consist of transmitting the “Start Transmission” sequence, followed by the command code, then the reception or transmission of the command parameter (if it exists) and finally the reception of the control sum CRC8 (if generated by the sensor).

If the serial communication between the microcontroller and the sensor is lost, then it can be re-established with a reset connection sequence. This consists of keeping the data line on logical level “1” during which at least nine periods of clock are sent, then a “Start Transmission” (Fig. 7). In this case the serial interface of the sensor is reset, while the status registry is not.

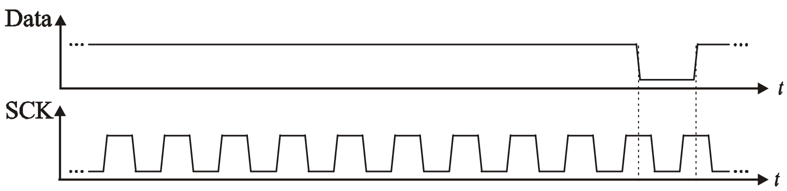

Fig. 7.

The checksum CRC8 is in fact Cyclic Redundancy Check using the generating polynomial $g(x) = x^8 \oplus x^5 \oplus x^4 \oplus 1$ and it is calculated for all the data in a command. If a discrepancy is detected between the checksum generated by the sensor and the one calculated by the microcontroller, then the sensor must be reset by software and the measurement should be restarted. If the two sums coincide, then the data bytes are received correctly and they represent the numeric value of the conducted measurement.

The command program consists of a segment of initializations which load the application variables with the necessary numerical values, programs the serial asynchronous interface, *T*1 counter in order to set the data transfer rate to 9600 bps, *T*0 counter to generate periodic interrupts, checks the operation status of the serial console and displays an application launch message. Then a basic program is executed, consisting of a keyboard query loop, during which various user commands can be read, followed by the corresponding program sequences.

The significance of the main commands implemented through software for this application is described below.

Command (A.) for powering the sensor in DC. The command produces the connection ($\overline{CA}=0$) / disconnection ($\overline{CA}=1$) of the power source. After connecting the power supply, it follows a one-second delay, then the transmission of reset connection sequence and “Start Transmission”. The RETURN (.) terminator is used for closing all commands.

The command (C.) displays a list with all the application’s commands.

The software reset command (SR.) initializes the serial interface from the sensor and loads the status register with the default value (00H).

The connection reset command (CR.) initializes the sensor serial interface, but maintains the content of the status register.

The status register read command (RSR.) displays the contents at the console.

The status register write command (WSR p.) reads from the console the numerical value that will be subsequently loaded into it.

The relative humidity measuring command reads the data bytes containing the measured numerical value and displays at the console. Three ways of displaying the relative humidity are implemented: hexadecimal (RHH.), binary (RHB.) and decimal (RHZ.).

The temperature measuring command reads the bytes of data containing the numerical values and displays at the console. There are also three ways of displaying the temperature: hexadecimal (TH.), binary (TB.) and decimal (TZ.). Using the application sensor, for the two last commands the following numerical values have been measured ($SR_0 = 0$):

$$RH_{ADC} = 489H \quad (RH_{ADC} = 010010001001B,\ RH_{ADC} = 1{,}161);$$
$$T_{ADC} = 17CFH \quad (T_{ADC} = 01011111001111B,\ T_{ADC} = 6{,}095).$$

The compensation and transformation of relative humidity command (RHL.) read from the sensor, into a numerical value with a physical significance is calculated with the expression (1), converted to decimal and truncated

$$RH_{Lin} = 39.2\%.$$

The relative humidity compensation command (RHT.) with temperature variation is calculated using expression (2), converted to decimal and truncated

$$RH_T = 38.9\%.$$

The compensation and transformation of temperature command (TCD.) read from the sensor, into Celsius degrees is calculated with the expression (3), converted to decimal and truncated

$$T_{°C} = 21.3°C.$$

The recording command (I.) reads the measured relative humidity and temperature and saves all the numeric results into an Excel file that is sent to the personal computer over the serial RS232 interface.

The load command (L.) of the internal EEPROM data memory of the microcontroller or of an output storage device, stores the numerical values of the relative humidity and the temperature, measured at certain time intervals. After a preset number of records, an Excel file is formed, that is sent to a personal computer.

Other command variants may calculate the average of the relative humidity and the temperature over a time interval, determine the minimal or maximum value for either of these measured parameters, monitor the relative humidity and the temperature in order to send certain commands when specified thresholds are reached etc.

5. Conclusions

The hardware structure previously described was built in practice and consists of the digital relative humidity sensor SHT75 without need for external components and a development system equipped with the AT89S8253 microcontroller. The selected sensor assures a high quality of measured signals, a quick response time, resilience to external interferences and a competitive price. It is used in relative humidity and temperature measuring activities, in command and control regulator systems for industrial or thermal processes, in meteorological stations, in health monitoring systems, etc.

The command program has a series of commands and subroutines implemented which help achieve the measurement of relative humidity and temperature, sensor compensating, transforming data from the sensor into numerical values, compensating relative humidity with temperature, showing data on a console in various formats, the conversion of the results into decimal, the insertion of the measured values into Excel files and sending them to a personal computer for more complex operations, etc.

The program is written in machine code, it occupies an amount of 3.5 kb of memory and it stands out through its small volume of the required memory. The commands written for measuring relative humidity and temperature are provided with a subroutine which calculates and verifies the checksum in order to establish if the data have been correctly received.

The measuring and monitoring of this data allows the implementation of a software regulator with various programmable thresholds which can send multiple commands to the monitored process. The existence of the two sensors in the same chip allows for the calculation of the dew point when needed.

REFERENCES

Duma P., *Microcontrolerul INTEL 8051. Aplicații*. Edit. „TEHNOPRESS”, Iaşi, 2004.
Hintz J.K., Tabak D., *Microcontrollers. Architecture, Implementation and Programming*. McGraw Hill, New York, 1993.
Lance A. *Leventhal, Programmation en langage assembleur.* Edit. Radio, Paris, 1998.
Peatmann B.J., *Design with Microcontrollers*, McGraw Hill, New York, 1998.
* * * *ATMEL, Family Microcontroller,* Data Book, 1998.
* * * *ATMEL, AT89S8253 Microcontroller,* Data Sheet, 2005.
* * * *MAXIM, Multichannel RS232 Drivers/Receivers*. MAX232A Data Sheet, 2006.
* * * *Sensirion, Humidity and Temperature Sensor*. SHT75 Data Sheet, version V3, 2007.
* * * *Sensirion, Status Register*. Application Note, 2006.
* * * *Sensirion, Non-Linearity Compensation*. Application Note, 2008.
* * * *Sensirion, Dew Point Calculation*. Application Note, 2008.
* * * *Sensirion, CRC Checksum Calculation*, Application Note, 2011.
* * * *Sensirion, Humidity at a Glance. Most Relevant Equations with Sample Code*. 2008.
* * * *Sensirion, Introduction to Humidity. Basic Principles on Physics of Water Vapor*. 2009.
* * * *Sensirion, Conditions of Thermal Comfort. Influence of Humidity and Temperature on Personal Wellbeing*. 2010.
* * * *Taiwan Semiconductor*. TS1117 Data Sheet, 2003.
* * * *Texas Instruments*. Data Book, 1992.

MONITORIZAREA SENZORULUI DE UMIDITATE RELATIVĂ SHT75 CU MICROCONTROLER DIN FAMILIA ATMEL

(Rezumat)

Se descrie interfața hardware necesară pentru măsurarea umidității relative şi a temperaturii, utilizând senzorul digital de umiditate SHT75 şi un sistem de dezvoltare echipat cu microcontroler din familia ATMEL. Programul de comandă realizat măsoară şi afişează umiditatea relativă şi temperatura, compensează mărimile măsurate, transformă datele citite din senzor în valori numerice cu semnificație fizică, transmite la un calculator personal datele achiziționate şi efectuează diverse prelucrări cu mărimile măsurate.