

BULETINUL INSTITUTULUI POLITEHNIC DIN IAȘI  
Publicat de  
Universitatea Tehnică „Gheorghe Asachi” din Iași  
Tomul LIX (LXIII), Fasc. 1, 2013  
Secția  
ELECTROTEHNICĂ. ENERGETICĂ. ELECTRONICĂ

## MONITORING THE MPL115A1 PRESSURE SENSOR USING A ATMEL FAMILY MICROCONTROLLER

BY

PETRUȚ DUMA<sup>1,\*</sup>, EUGEN PETAC<sup>2</sup> and ABDEL RAHMAN ALZOUBAIDI<sup>3</sup>

<sup>1</sup>“Gheorghe Asachi” Technical University of Iași  
Faculty of Electronics, Telecommunications and Information Technology

<sup>2</sup>“Ovidius” University of Constanța,

Faculty of Mathematics and Computer Science

<sup>3</sup>Al Balqa Applied University (APU), Kingdom of Jordan,  
Department of Computer Engineering

Received: March 4, 2013

Accepted for publication: March 26, 2013

**Abstract.** The paper describes the hardware interface required for measuring pressure and temperature using the digital pressure sensor MPL115A1 and a development system equipped with an ATMEL family microcontroller. The command and control of this sensor are made by a user through a personal computer connected to the serial RS232 interface of the microcontroller. The command program measures and displays the pressure and the temperature, calculates the compensated pressure, performs various processing of the measured values.

**Key words:** digital pressure sensor; ATMEL microcontroller; hardware interface; application system; command program.

### 1. Introduction

The pressure sensor, such as other transducers, *e.g.* those for humidity, 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. In the end, using

---

\*Corresponding author: *e-mail*: pduma@etti.tuiasi.ro

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 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 pressure monitoring system, used in fact for monitoring also other physical properties such as humidity, 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 meanings: PS – pressure sensor, PSI – pressure sensor interface, HS – humidity sensor, HIS – humidity sensor interface, TS – temperature sensor, TSI – temperature sensor interface,  $\mu\text{C\_AS}$  – 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.

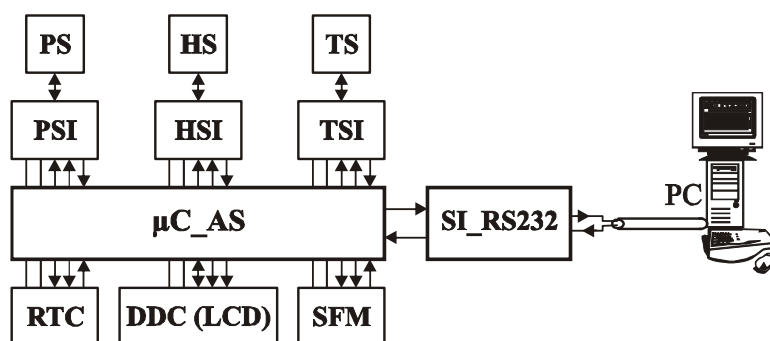


Fig. 1

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 pressure sensor monitoring. Further papers will be drawn to describe the humidity and temperature sensor monitoring, the real-time clock and the serial FLASH management.

## 2. The MPL115A1 Digital Pressure Sensor

The main features of this digital pressure sensor, manufactured by Freescale Semiconductor, are: the absolute pressure measuring range from 50 kPa to 115 kPa;  $\pm 1$  kPa accuracy; 0.15 kPa resolution; digitized values for uncompensated pressure and temperature values are 10-bit long; the serial peripheral interface used to communicate with the microcontroller has a 8 Mbit/s throughput; no external components required; the operating temperature range between  $-40^{\circ}\text{C}$  and  $+105^{\circ}\text{C}$ , allowing the device to operate in harsh environment; each chip is individually factory calibrated and the required coefficients are stored in a nonvolatile memory; the pressure and temperature data outputs are monotonous functions.

The internal block schematic of the MPL115A1 pressure sensor is represented in Fig. 2. The notes have the following meaning: PMB – pressure measure bridge, DMA – differential measure amplifier, TS – temperature sensor, AM – analogue multiplexer, ADC – analogue–digital converter, DDM – digital demultiplexer, PR – pressure register, TR – temperature register, NVM – nonvolatile memory for calibration coefficients ( $a_0$ ,  $b_1$ ,  $b_2$ ,  $c_{12}$ ), AD – address decoder, CSR – conversion start register, SB – supply block, CCL – command and control logic, SPI – serial peripheral interface.

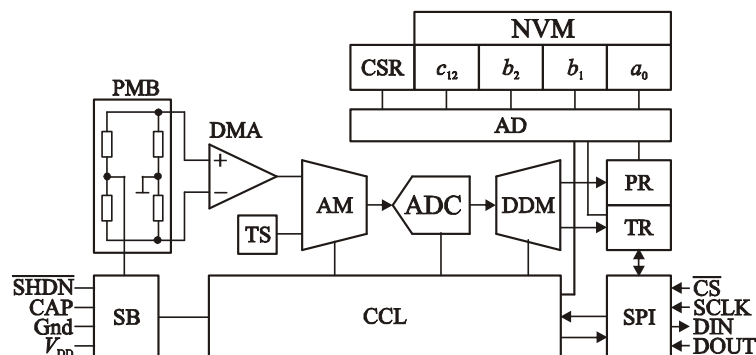


Fig. 2

The pressure sensor uses a piezoresistive technology consisting of a micro-processed silicon diaphragm and a piezoelectric strain gauge. When pressure is applied on the sensor, the diaphragm is strained and that produces a variation in the resistance of the piezoresistive layer. In order to measure this variation, an electrical current is applied to this layer, connected into a measuring bridge, thus generating a voltage proportional with the pressure applied. In order to measure the pressure, the obtained signal must be amplified, then converted from analogue to digital, both for pressure and temperature, and the resulting data is stored in the corresponding registers.

The master microcontroller manages the slave pressure, using SPI interface to transmit the start conversion command and to read the raw measured values of pressure and temperature and the calibration coefficients. After running a compensation algorithm, the compensated value of the absolute atmospheric pressure is obtained.

After the sensor is powered, the microcontroller is able to transmit the required commands for reading the coefficients necessary for the compensation. In this case, the main circuits of the sensor are disabled, and the coefficients stored in the nonvolatile memory are transferred into the microcontroller's memory. The coefficients may be re-read at any point, while their values remain constant for the same device. Obviously, these values differ from one device to another.

Sending the CONVERT command to the pressure sensor accesses the conversion start register that enables the main circuits of the device and starts the measuring process of the absolute pressure and, afterwards, of the temperature. The results of these two successive conversions are loaded in the corresponding registers and then the sensor goes into stand-by mode. The conversion time,  $t_c$ , is typically 1.6 ms and can get up to a maximum of 3 ms. After the conversion is ended, the microcontroller sends the commands for reading the most significant byte (MSB) and, respectively, the least significant byte (LSB) for both the measured absolute pressure and temperature. The measured pressure,  $P_{ADC}$ , and the measured temperature,  $T_{ADC}$ , are both unsigned 10-bit integers ( $I_9I_8I_7I_6I_5I_4I_3I_2I_1I_0$ , where  $I_k$  is the rank  $k$  bit of the respective property ( $k = 0, \dots, 9$ )). The structure of the two bytes of the pressure and of the temperature registers, has the following format

$$\begin{aligned} P_{ADC\_MSB} &= I_9I_8I_7I_6I_5I_4I_3I_2, \\ P_{ADC\_LSB} &= I_1I_0000000, \\ T_{ADC\_MSB} &= I_9I_8I_7I_6I_5I_4I_3I_2, \\ T_{ADC\_LSB} &= I_1I_0000000. \end{aligned}$$

The digital values of the measured pressure and temperature are

$$\begin{aligned} P_{ADC} &= I_9I_8I_7I_6I_5I_4I_3I_2I_1I_0, \\ T_{ADC} &= I_9I_8I_7I_6I_5I_4I_3I_2I_1I_0. \end{aligned}$$

All the coefficients used for compensating the measured pressure are represented in 2's complement, the most significant bit indicating the sign. The main properties of each coefficient are briefly presented in the following section, along with the structure of the two bytes (MSB and LSB) stored in the nonvolatile memory of the sensor and the corresponding digital value.

$a_0$  is the pressure offset coefficient, 16-bit long, out of which the first bit stores the sign ( $S$ ), the next 12 bits are used for the integer part ( $I_{11}I_{10}I_9I_8I_7I_6I_5I_4I_3I_2I_1I_0$ ) and the last 3 bits are used for the fractional part ( $F_2F_1F_0$ , where  $F_k$  is the rank  $k$  fraction bit of the respective coefficient, ( $k = 0, \dots, 2$ )):

$$\begin{aligned} a_{0\_MSB} &= SI_{11}I_{10}I_9I_8I_7I_6I_5, \\ a_{0\_LSB} &= I_4I_3I_2I_1I_0F_2F_1F_0, \\ a_0 &= SI_{11}I_{10}I_9I_8I_7I_6I_5I_4I_3I_2I_1I_0 \bullet F_2F_1F_0. \end{aligned}$$

$b_1$  is the pressure sensitivity coefficient, 16-bit long: first bit is the sign bit ( $S$ ), 2 bits are used to store the integer part ( $I_1I_0$ ) and 13 bits for the fractional part ( $F_{12}F_{11}F_{10}F_9F_8F_7F_6F_5F_4F_3F_2F_1F_0$ ):

$$b_{1\_MSB} = SI_1I_0F_{12}F_{11}F_{10}F_9F_8,$$

$$b_{1\_LSB} = F_7F_6F_5F_4F_3F_2F_1F_0,$$

$$b_1 = SI_1I_0 \cdot F_{12}F_{11}F_{10}F_9F_8F_7F_6F_5F_4F_3F_2F_1F_0.$$

$b_2$  is the temperature offset coefficient, 16-bit long: first bit is the sign bit ( $S$ ), one bit is used to store the integer part ( $I_0$ ) and 14 bits for the fractional part ( $F_{13}F_{12}F_{11}F_{10}F_9F_8F_7F_6F_5F_4F_3F_2F_1F_0$ ):

$$b_{2\_MSB} = SI_0F_{13}F_{12}F_{11}F_{10}F_9F_8,$$

$$b_{2\_LSB} = F_7F_6F_5F_4F_3F_2F_1F_0,$$

$$b_2 = SI_0 \cdot F_{13}F_{12}F_{11}F_{10}F_9F_8F_7F_6F_5F_4F_3F_2F_1F_0.$$

$c_{12}$  is the temperature sensitivity coefficient, 14-bit long: first bit is the sign bit ( $S$ ), no bit used for the integer part and 13 bits for the fractional part ( $F_{13}F_{12}F_{11}F_{10}F_9F_8F_7F_6F_5F_4F_3F_2F_1F_0$ ). It also includes 9 bits set to logical "0" inserted between the decimal point and the fractional part:

$$c_{12\_MSB} = SF_{12}F_{11}F_{10}F_9F_8F_7F_6,$$

$$c_{12\_LSB} = F_5F_4F_3F_2F_1F_000,$$

$$c_{12} = S 0.000000000 F_{12}F_{11}F_{10}F_9F_8F_7F_6F_5F_4F_3F_2F_1F_0.$$

The absolute compensated pressure,  $P_{COMP}$ , is calculated using the relation

$$P_{COMP} = a_0 + (b_1 + c_{12} \cdot T_{ADC}) \cdot P_{ADC} + b_2 \cdot T_{ADC}. \quad (1)$$

This relation results in a digital value of 0 for an applied pressure of 50 kPa and an end-of-scale value of 3 FFH corresponding to a measured pressure of 115 kPa. In the end, this result is converted to decimal and the absolute compensated pressure is calculated, in kPa, using the relation

$$P = 50 + \frac{115 - 50}{1023} \cdot P_{COMP}. \quad (2)$$

### 3. Interfacing the Pressure Sensor MPL115A1

The command and control of the digital pressure sensor MPL115A1 that was used in the first testing, checking and development stage of the application, are made using a development system equipped with microcontroller AT89S8253. Fig. 3 presents the simple interface for pressure sensor monitoring, that uses port P1 of the microcontroller.

The pressure sensor is powered in the voltage range from 2.375 V minimum to 5.5 V maximum, but typically a 3.3 V voltage is used. In the hardware interface designed, the sensor is powered at 3.3 V from the integrated three-point positive voltage stabilizer TS1117-3V3, which requires only

decoupling capacitors as external components. The command of the DC power supply is made through an open collector buffer and a transistor  $T$  (BC556). The signal  $\overline{CA}$  assigned to this command, active on logical “0”, is software provided software on line  $P1.5$  of the microcontroller. The internal supply block of the sensor has an external connection, CAP, that requires an external ceramic capacitor of  $1\ \mu\text{F}$  for decoupling the output of the main internal stabilizer.

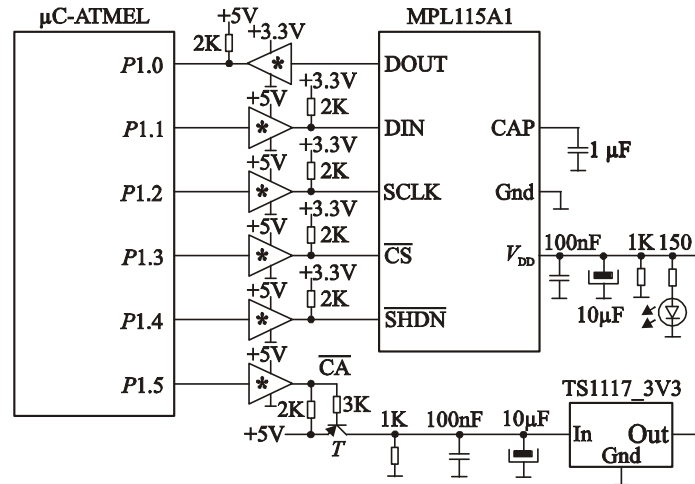


Fig. 3

After powering the sensor, the device waits for the power voltage to stabilize, then all the blocks become active, the serial peripheral interface terminals go into a high impedance state, the internal registers are reset, then the device passes into stand-by mode. The current consumption of the sensor during operation (while receiving commands, measuring pressure and temperature, sends coefficients and data) is of  $20\ \mu\text{A}$  maximum, while in stand-by goes below  $10\ \mu\text{A}$ . During longer inactivity periods, in order to reduce the energy consumption, the pressure sensor can be set into low power mode (shutdown) by connecting the input  $\overline{SHDN}$  to logical “0”. In this situation, the power supply of the internal circuits is cut, the serial interface is not able to communicate with the microcontroller, while the current input goes down to  $1\ \mu\text{A}$ , which is a major advantage for applications powered by batteries. This state is abandoned by connecting input  $\overline{SHDN}$  to logical “1”, then, after waiting for a delay,  $t_w$ , of maximum 5 ms, the device becomes operational but with the internal registers reset.

The serial data output, DOUT, the serial data input, DIN, the clock serial input, SCLK, the circuit selection input,  $\overline{CS}$ , and the input for low power operating mode (shutdown),  $\overline{SHDN}$ , are connected through open collector buffers to the lines of the parallel port  $P1$  (Fig.3). These buffers are useful for

translating the logical levels between the microcontroller from the development system powered at +5 V and the pressure sensor powered at 3.3 V. The final version of the hardware structure of this application will be managed by an application system equipped with a microcontroller powered at 3.3 V, and these buffers will no longer be required.

#### 4. Monitoring the Pressure Sensor MPL115A1

The data transfers between the microcontroller and the pressure sensor are made by setting the selection signal,  $\overline{CS}$ , to logical “0” (if the system is powered  $\overline{CA} = 0$ , the low power operation mode is not selected,  $\overline{SHDN} = 1$ , and the clock signal SCLK is set to logical “0”). Based on the clock signal provided by the microcontroller, data are written on the rising edge, and are read by the sensor on the falling edge. If the  $\overline{CS}$  signal becomes logical “1”, any communication between the microcontroller and the MPL115A1 circuit is ended. Data communication is made starting with the most significant bit in the first position and ending with the least significant bit in the last one.

Any command is initiated by the master microcontroller and consists of sending to the slave pressure sensor an 8-bit command word in the first part, followed by an 8-bit data that is read or written in the second part. All the reading and writing operations are made with the nonvolatile memory locations where the compensation coefficients are stored, with the registers containing the measured value of the pressure and of the temperature, or with the conversion start register. These commands constitute a way of addressing memory locations in order to read or write data. Table 1 lists all the commands used for the pressure sensor MPL1511A1.

**Table 1**

Name	$\overline{R/W}$	Address	Commands	Description
$P_{ADC\_MSB}$	1	00H	10000000B = 80H	Read MSB measured pressure
$P_{ADC\_LSB}$	1	01H	10000010B = 82H	Read LSB measured pressure
$T_{ADC\_MSB}$	1	02H	10000100B = 84H	Read MSB measured temperature
$T_{ADC\_LSB}$	1	03H	10000110B = 86H	Read LSB measured temperature
$a_0\_MSB$	1	04H	10001000B = 88H	Read MSB $a_0$ coefficient
$a_0\_LSB$	1	05H	10001010B = 8AH	Read LSB $a_0$ coefficient
$b_1\_MSB$	1	06H	10001100B = 8CH	Read MSB $b_1$ coefficient
$b_1\_LSB$	1	07H	10001110B = 8EH	Read LSB $b_1$ coefficient
$b_2\_MSB$	1	08H	10010000B = 90H	Read MSB $b_2$ coefficient
$b_2\_LSB$	1	09H	10010010B = 92H	Read LSB $b_2$ coefficient
$c_{12\_MSB}$	1	0AH	10010100B = 94H	Read MSB $c_{12}$ coefficient
$c_{12\_LSB}$	1	0BH	10010110B = 96H	Read LSB $c_{12}$ coefficient
Convert	0	12H	00100100B = 24H	Write start conversion

The 8-bit command words consist of

a)  $R/\overline{W}$  – 1 bit read/write identifier. It has the logical value “1” for a read operation and “0” for write. This is the most significant bit and it is sent always the first;

b)  $A_5-A_0$  – the 6-bit address of the memory location or of the register used;

c)  $X$  – the least significant bit. It is not used and can have any value whatsoever. In this application,  $X = 0$ .

The evolution of the signals from the serial peripheral interface used to command and control the pressure sensor, during the data read/write operations, are presented in Figs. 4 and 5, respectively.

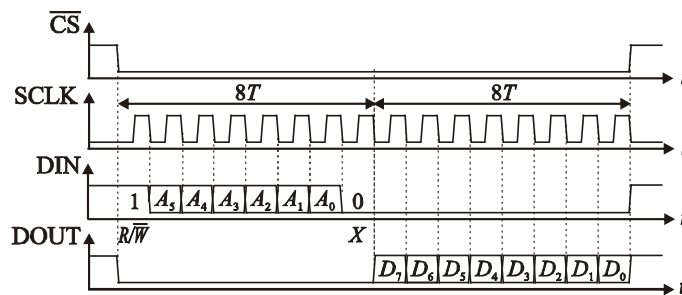


Fig. 4

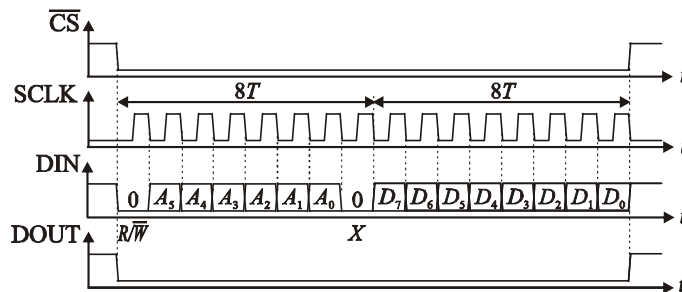


Fig. 5

The diagrams for reading the compensation coefficients from the nonvolatile memory of the sensor, for performing the start conversion command and for reading the measured pressure and temperature, are shown in Figs. 6, 7 and 8, respectively.

The command program consists of an initialization sequence that loads the variables with the required digital numerical, erases the console screen, then displays an execution launch message. Afterwards, a keyboard interrogation loop is entered, during which different user commands are sent. The signification of the main software implemented commands is briefly described below.



The power-on command, (A.), determines the pressure sensor to connect/disconnect to the DC power supply.

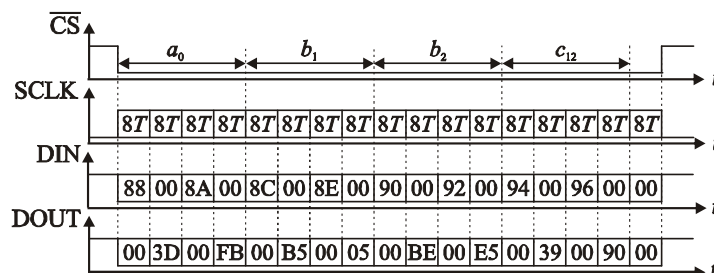


Fig. 6

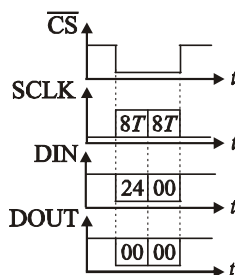


Fig. 7

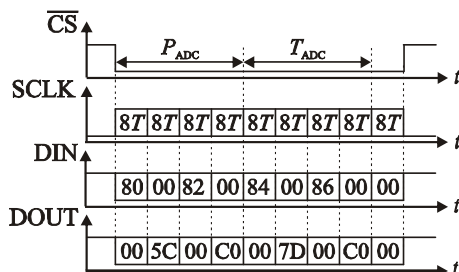


Fig. 8

The low power device operation mode command, (S.), sets the device in the low power mode or resumes from this operation mode.

The coefficient read command, (C), gets the coefficients used in the compensation process ( $a_0$ ,  $b_1$ ,  $b_2$  and  $c_{12}$ ), stores them in the application's variable area and displays them on the console. Three software versions are implemented: CH command for displaying the coefficients in hexadecimal (MSB and LSB), CB command for binary display and CZ for decimal display. For the sensor used in this particular application, the use of these commands have produced the following values:

CH.

$$\begin{aligned} a_0 &= 3DFBH, \\ b_1 &= 0B505H, \\ b_2 &= 0BEE5H, \\ c_{12} &= 3990H. \end{aligned}$$

CB.

$$\begin{aligned} a_0 &= + 11110111111.011B, \\ b_1 &= - 10.0101011111011B, \\ b_2 &= - 1.00000100011011B, \\ c_{12} &= + 0.0000000001110011001B. \end{aligned}$$

CZ.

$$\begin{aligned} a_0 &= + 1983.375, \\ b_1 &= - 2.3431396484, \\ b_2 &= - 1.01727294921875, \\ c_{12} &= + 0.000878334045410156. \end{aligned}$$

The start conversion command (*C.*) sends to the pressure sensor the command CONVERT, that performs the measurement of the absolute atmospheric pressure and of the temperature.

The read command, (*R.*), gets the measured pressure and temperature, stores them in the application variables area and displays them on the console. In this case there were also implemented three software versions: command RH used to display the pressure and the temperature in hexadecimal, command RB for binary display and command RZ for decimal display. The following numeric values were determined during an average day:

RH.

$$\begin{aligned} P_{ADC} &= 5C80H, \\ T_{ADC} &= 7DC0H. \end{aligned}$$

RB.

$$\begin{aligned} P_{ADC} &= 101110010B, \\ T_{ADC} &= 111110111BH. \end{aligned}$$

RZ.

$$\begin{aligned} P_{ADC} &= 370, \\ T_{ADC} &= 503. \end{aligned}$$

The command used to calculate the compensated pressure (*P.*) calls a few arithmetic subroutines in order to compute this value using relation (1) and then performs the conversion of the pressure to kPa using eq. (2), further converts the result to decimal and trims the result to one decimal point. The numerical value of the measured pressure determined in this application is

P.

$$P = 98.8 \text{ [kPa]}.$$

The recording command (*I.*) reads the compensation parameters, sends the start conversion command, reads the measured pressure and temperature and saves all the numerical results into an Excel file that is send to the PC over the serial RS232 interface.

The load command, (*L.*), for the external RAM memory of the microcontroller equipped system, stores the numerical values of the pressure 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 pressure and the temperature over a time interval, determine the minimal or maximal value for either of these parameters or monitor the pressure and the temperature in order to send certain commands when specified thresholds are reached.

## 5. Conclusions

The described hardware structure was built in practice; it is simple and consists of a digital pressure sensor MPL115A1 that does not require external components and a development system equipped with microcontroller AT89S8253. It is used in applications for pressure and temperature measurement, for industrial or thermal system control using regulators, for height measurement in meteorological stations, in health monitoring systems and so on.

The command program that was written implements a command and subroutines set, allowing the user to perform the reading of the compensation coefficients, to launch the start conversion command, to read the measured pressure and temperature, to calculate the compensated pressure and to convert it to decimal, to display the results on the console in various formats, to save the read and measured values in Excel file records and to send them to a personal computer for further processing, etc. This command program uses 3.2 kB of program memory area and it remarks by this low amount of required memory compared to the possibilities it offers and the features implemented.

Measuring and monitoring pressure and temperature allow the implementation of a software regulator with different pressure and temperature thresholds, that transmits multiple commands to the monitored process, but performs also other calculations for averaging, detecting the extreme values from a data set and so forth.

## REFERENCES

- Duma P., *Microcontrolerul INTEL 8051. Aplicații*. Edit. TEHNOPRESS, Iași, 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.  
\* \* \* *DALLAS Semiconductor*. Data Book, 2003.  
\* \* \* *Freescale Semiconductor, How to Implement the Freescale MPL115A1 Digital Barometer*. Application Note, 2009.  
\* \* \* *Freescale Semiconductor, MPL115A1 Data Sheet*. 2011.

- \* \* \* *MAXIM, Multichannel RS232 Drivers/Receivers. MAX232A Data Sheet, 2006.*
- \* \* \* Taiwan Semiconductor, TS1117 Data Sheet, 2003.
- \* \* \* Texas Instruments, Data Book, 1992.

## MONITORIZAREA SENZORULUI DE PRESIUNE MPL115A1 CU MICROCONTROLLER DIN FAMILIA ATMEL

(Rezumat)

Se descrie interfața hardware necesară pentru măsurarea presiunii și a temperaturii utilizând senzorul digital de presiune MPL115A1 și un sistem de dezvoltare echipat cu microcontroler din familia ATMEL. Comanda și controlul acestui senzor se realizează prin intermediul unui calculator personal conectat la interfața serială RS232 a microcontrolerului. Programul de comandă măsoară și afișează presiunea și temperatura, calculează presiunea compensată, efectuează diverse prelucrări cu mărimile măsurate.