

## TEMPERATURE SENSOR RESISTANCE CONVERSION TO BINARY CODE USING PULSE WIDTH MODULATION

**Aleksandr Vosstrukhin<sup>1</sup>, Elena Vakhtina<sup>2</sup>**

<sup>1</sup>Stavropol Technological Institute of Service, Russia; <sup>2</sup>Stavropol State Agrarian University, Russia  
 avosstrukhin@yandex.ru, eavakhtina@yandex.ru

**Abstract.** The presented researches are related to measurement technologies and can be used to design intelligent resistive sensors. To convert the resistance of resistive sensors into binary code in microprocessor-based systems, bridge circuits are used that interface with analog-to-digital converters (ADC). The bridge circuit of a resistive sensor is the main circuit solution for cases where the task is to ensure high measurement accuracy. However, the problem of eliminating the nonlinearity of the conversion characteristic remains and needs to be solved. As a rule, nonlinearity is eliminated programmatically using conversion tables or selection of an analytical approximating function, which is not the best option. In this paper, the linearization problem of the conversion characteristic is solved by automatically balancing the bridge circuit using the pulse-width modulation method. In some cases, it is necessary to use amplifiers to interface bridge circuits with ADC. These elements introduce additional errors in the conversion result, as well as complicate the hardware. A variant of the circuit solution without ADC and amplifier is proposed. The paper discusses a microcontroller measuring converter (MMC), which allows obtaining a linear dependence of the binary code on the resistance of the temperature sensor. The equations for calculating software-realizable parameters are given, which provide the required resistance resolution. The MMC resolution is 1 Ω, when the sensor resistance changes from 500 to 1500 Ω at a measuring current of 1.23 mA. In this case, MMC conversion time – 4 ms. Experimental studies were performed on the basis of the Atmel's 8-bit AVR microcontroller. The study results can be applied to design intelligent sensors using other resistive sensors, such as strain gages.

**Keywords:** analog comparator, fill factor, microcontroller, measuring current, resolution.

### Introduction

Resistive sensors are most prevalent in monitoring and control systems. They measure such values as: temperature, weight, pressure, humidity, gas flow rate, etc. The operation of resistive sensors is based on the dependence of the sensitive element resistance on the measured physical quantity. For example, when measuring temperature, the dependence of the platinum wire resistance on temperature is used.

The choice of the temperature sensor should be made taking into account a number of parameters. When searching for the optimal sensor, in the first place such qualities are put, as power consumption, dimensions, ease of connection, possibility of parallel operation in a network configuration, and so on. There is no universal sensor suitable for all cases. Developers have to look for compromise options.

The most universal sensor will be the one that functions under the control of a programmable system, for example, a microcontroller. The functionality and the process of improving such a sensor are determined mainly by the improvement of its software. For such a sensor, a software library can be created, with a set of functions from which you can choose the one you need for a particular case. Such sensors are called intelligent [1].

This paper discusses a microcontroller measuring converter (MMC) [2], which can be used to build intelligent resistive sensors.

Table 1 shows the main technical characteristics of known digital temperature sensors for comparison with the result obtained in our researches.

**Table 1**  
**Main technical characteristics of digital temperature sensors**

Digital temperature sensors	Operating temperature range, °C	Maximum resolution, °C	Consumed current (modes: active / sleep)	Conversion time, ms
DS18B20 [3]	-55 ... 125	0.0625	1.5 mA / 1 µA	750
STLM75 [4]	-55 ... 125	0.5	150 µA / 1 µA	150
TMP107 [5]	-55 ... 125	0.0156	400 µA / 10 µA	18
LMT01 [6]	-50 ... 150	0.0625	40 µA / 3.5 µA	100

## Materials and methods

The process of converting resistance into a binary code consists of two transformations: resistance into voltage, and then this voltage into binary code.

Analog-to-digital converters (ADC) are used to convert voltage to binary code. To effectively pair the sensors with the ADC, an amplifier is required. In some cases, measuring circuits, such as a bridge circuit, can be connected directly to the ADC input. But, at the same time, tasks arise in coordinating the range of the change of the controlled magnitude with the ADC. To solve the matching task, instrumental (measuring) amplifiers are often used [1]. Amplifiers and ADCs introduce additional errors in the conversion result, as well as complicate the hardware. An alternative variant of converting resistance to binary code without using an amplifier and ADC is considered in this work.

The functional diagram for the MMC of resistance of a resistive sensor into a binary code is shown in Fig. 1. MMC includes the following elements: a microcontroller (MCU), which contains pulse-width modulators PWM1, PWM2 and an analog comparator (AC); RC-filters F<sub>1</sub> and F<sub>2</sub>; a measuring circuit containing a resistive sensor R<sub>C</sub> and a reference resistor R<sub>r</sub>; a reference circuit containing resistors R<sub>1</sub>, R<sub>2</sub> [7, 8].

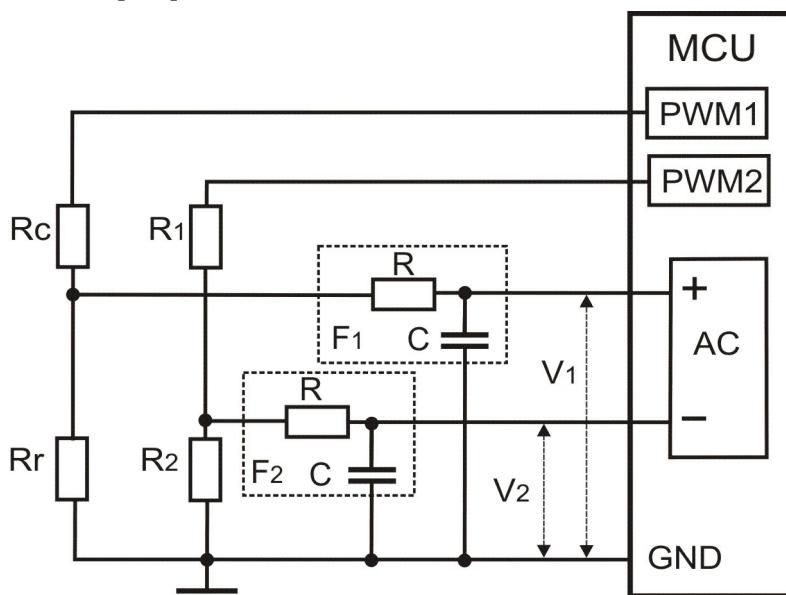


Fig. 1. MMC functional diagram

Consider the basic relationships that allow establishing the necessary dependencies. An average voltage of the measuring circuit is defined by equation (1):

$$V_{PWM1} = V_H \cdot G_1, \quad (1)$$

where  $V_H$  – high level voltage (logical “1”);

$G_1$  – fill factor of the PWM-signal generated by the PWM1.

We assume  $V_H$  equal to  $V_{CC}$  (voltage of the microcontroller power source), then the equation (1) takes the form (2):

$$V_{PWM1} = V_{CC} \cdot G_1 \quad (2)$$

Actually, the voltage is  $V_H \approx 4.95$  V at a current through the output  $I_{OH} \approx 2$  mA;  $V_{CC} = 5$  V and the microcontroller temperature  $t = 25$  °C [9].

The positive Analog Comparator pin has the voltage  $V_1$ , defined by formula (3):

$$V_1 = V_{PWM1} \cdot \frac{R_r}{R_C + R_r}. \quad (3)$$

Taking into account formulas (2) and (3), we obtain (4):

$$V_1 = V_{CC} \cdot G_1 \cdot \frac{R_r}{R_C + R_r}. \quad (4)$$

The average voltage of the reference circuit is determined by formula (5), which is similar to (1):

$$V_{PWM2} = V_H \cdot G_2, \quad (5)$$

where  $G_2$  – fill factor of the PWM-signal generated by the PWM2.

Then, analogously to formula (2), we obtain (6).

$$V_{PWM2} = V_{CC} \cdot G_2, \quad (6)$$

The negative Analog Comparator pin has the voltage  $V_2$ , defined by formula (7):

$$V_2 = V_{PWM2} \cdot \frac{R_2}{R_1 + R_2}. \quad (7)$$

Taking into account equations (6) and (7), we obtain (8):

$$V_2 = V_{CC} \cdot G_2 \cdot \frac{R_2}{R_1 + R_2}. \quad (8)$$

The MMC operating algorithm is as follows. In the initial state  $R_C = R_r = R_1 = R_2$ , the microcontroller tunes PWM1 and PWM2 to form PWM-signals with duty factors  $G_1 = G_2$ . In this case  $V_1 \approx V_2$ . If  $V_1 > V_2$ , then “1” will be formed at the Analog Comparator output, if  $V_1 < V_2$ , then it will be “0”. The microcontroller continuously monitors the Analog Comparator output. If the voltage  $V_1$  becomes less than the voltage  $V_2$ , due to increasing  $R_C$ , the microcontroller increases  $G_1$ . In this case, in accordance with equation (4), the voltage  $V_1$  increases. As soon as the voltage  $V_1$  becomes higher than the voltage  $V_2$ , the microcontroller reduces  $G_1$ . Thus,  $V_1 \approx V_2$ .

The average current of the measuring circuit is determined according to (9):

$$I_C = \frac{V_{PWM1}}{R_C + R_r}. \quad (9)$$

Given equation (2), we obtain (10):

$$G_1 = \frac{V_{PWM1}}{V_{CC}} = \frac{I_C(R_C + R_r)}{V_{CC}}. \quad (10)$$

In equation (10) the values of  $V_{CC}$ ,  $R_r$ , and  $I_C$  are constants. To satisfy the condition  $V_1 = V_2$ , the current  $I_C$  is maintained by the microcontroller at a given level by balancing the resistive bridge  $R_r$ ,  $R_c$ ,  $R_1$ ,  $R_2$  [7]. Since the voltage  $V_2 = const$  and  $V_1 = const$ ,  $I_c = V_1/R_r = const$ . The resistance  $R_C$  changes only. Consequently, according to equations (9) and (10),  $G_1$  is a function of resistance  $R_C$ . Binary codes proportional to  $G_1$  and  $G_2$  are formed using the 16-bit timer/counter built into the microcontroller. The functioning of PWM is described in the source [9].

To implement MMC, it is advisable to use the “Fast PWM” mode. In this mode a PWM-signal of the highest frequency is generated, which allows using RC-filter elements with lower capacitance and resistance values. The frequency of the PWM-signal is determined by formula (11):

$$f_{PWM} = \frac{f_C}{TOP}, \quad (11)$$

where  $f_C$  – frequency of clock pulses arriving at the counting input of the timer/counter. In our case, it is equal to the frequency of the clock generator of the microcontroller, i.e.  $f_C = 16$  MHz;

$TOP$  – counting module; it can be arbitrary in the range from 3 to 65535, which corresponds to a resolution from 2 to 16 bits.

## Results and discussion

Experimental studies were performed in the “Fast PWM” mode and 12-bit resolution. The counting module in this case is  $TOP = 2^{12} = 4096$ . Then, according to equation (11) we get:

$$f_{PWM} = \frac{16 \cdot 10^6}{4096} = 3906 \text{ Hz}.$$

The fill factors  $G_1$  and  $G_2$  are directly proportional to the binary codes  $D_{OCR1}$  and  $D_{OCR2}$  loaded into the comparison registers OCR1 and OCR2, respectively PWM1 and PWM2.  $G_1$  and  $G_2$  are defined by equations (12):

$$G_1 = \frac{D_{OCR1}}{TOP}; G_2 = \frac{D_{OCR2}}{TOP}.$$

Considering formulas (10) and (12), we get (13) and (14): (12)

$$D_{OCR1} = \frac{I_C \cdot (R_C + R_r) \cdot TOP}{V_{CC}}. \quad (13)$$

$$D_{OCR2} = \frac{I_C \cdot (R_1 + R_2) \cdot TOP}{V_{CC}}. \quad (14)$$

In the initial state, when  $R_C = R_r = R_1 = R_2$ , the currents  $I_C$  of the measuring and  $I_r$  of the reference circuits are close in value, but not equal. The binary codes  $D_{OCR1}$  and  $D_{OCR2}$  are also close, but not equal. The conversion process is driven by this inequality. Only by mismatch between  $V_1$  and  $V_2$ , the microcontroller determines “to increment or decrement” the binary code  $D_{OCR1}$ .

PWM1 and PWM2-signals are formed using transistor switches. Transistor switches output resistances change as a result of heating from the current flowing through them and from the switching frequency. Since PWM1 and PWM2 operate at the same frequency, and the currents  $I_C$  and  $I_r$  flowing through them are close in value, then the output resistances of the transistor switches will be close also, and therefore the error of the MMC conversion will be minimal.

Experimental studies used the resistive sensor resistance range from 500 to 1500  $\Omega$ . This range is the most popular when measuring temperature by resistive sensors, such as platinum RTD 703-102BBB-A00 [10] and semiconductor KTY81/110 [11].

MMC resolution depends on  $I_C$ . To obtain the required  $I_C$ , it is required to calculate the initial binary code  $D_{OCR} = D_{OCR1} = D_{OCR2}$  using equations (13) or (14). The initial value of KTY81/110 sensor resistance  $R_C = 1000 \Omega$  for 25 °C will be taken [11]. As a result of the experiment, the dependence of  $D_{OCR}$  on the measuring current  $I_C$  for  $R_C = 1000 \Omega$  (Fig. 2) was established, confirming equations (13) and (14).

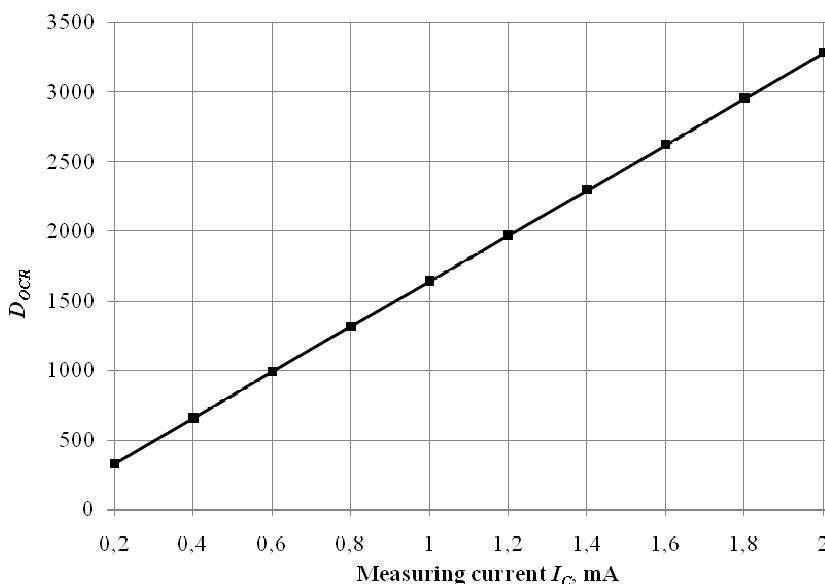


Fig. 2. Dependence of  $D_{OCR}$  on measuring current  $I_C$  for  $R_C = 1000 \Omega$

The MMC resistance resolution is determined by equation (15):

$$S_r = \Delta R_C / \Delta D, \quad (15)$$

where  $\Delta R_C = 1000 \Omega$  – sensor resistance change in the range from 500 to 1500  $\Omega$ ;  
 $\Delta D = 816$  – binary code change caused by the resistance change  $\Delta R_C$ , at  $I_C = 1$  mA.

Then  $S_r = 1000/816 = 1.23 \Omega$ .

The resistance of the KTY81/110 sensor varied from 490  $\Omega$  to 1696  $\Omega$ , when the temperature changed from -55 °C to 100 °C. As the  $R_C$ , a variable resistance box P33 was used. Fig. 3 shows the  $S_r$  resolution dependence on the measuring current  $I_C$ .

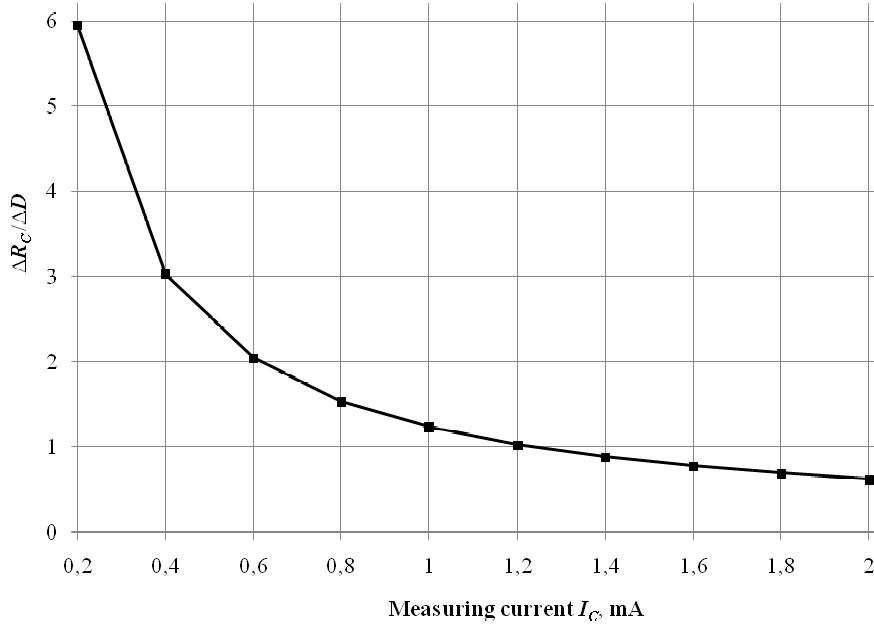


Fig. 3. Dependence of  $S_r$  resolution on  $I_C$  measuring current

Using the data of the KTY81/110 sensor [11] and Fig. 3, it is possible to determine the MMC temperature resolution  $S_t$  depending on  $I_C$ . For example, at  $I_C = 1$  mA:  $S_r \approx 1.23 \Omega$  and  $S_t \approx 0.15$  °C; at  $I_C = 2$  mA:  $S_r \approx 0.617 \Omega$  and  $S_t \approx 0.086$  °C. MMC spends 4 ms per one conversion. The maximum resolution of the DS18B20 sensor is 0.0625 °C, the conversion time takes 750 ms (Table 1). In this case, the MMC resolution is lower than DS18B20, but the conversion speed is  $750/4 \approx 188$  times higher. The MMC resolution can be increased by raising the  $I_C$  current.

Example: suppose you need to get the MMC resistance resolution  $S_r = 1 \Omega$ . To solve the task, we use the dependence shown in Fig 3. To ensure the specified resolution  $S_r = \Delta R_C / \Delta D = 1 \Omega$ , the current  $I_C \approx 1.23$  mA must flow through  $R_C$  (Fig.3). Determine the value of the binary code by formula (13):

$$D_{OCRI} = \frac{0.00123 \cdot (1000 + 1000) \cdot 4096}{5} \approx 2015.$$

We load the resulting binary code  $D_{OCRI} = 2015$  into the comparison registers PWM1 and PWM2. The obtained results are presented in Table 2.

Table 2  
Dependence of the binary code increment  $\Delta D$  on the resistance  $R_C$  at  $I_C = 1.23$  mA

$R_C, \Omega$	500	600	700	800	900	1000	1100	1200	1300	1400	1500
$\Delta D$	0	100	201	301	402	502	602	703	804	903	1003

According to Table 2, we obtained a linear dependence of  $\Delta D$  on  $R_C$  over the entire range of  $R_C$  variation. The resolution of the MMC is determined by formula (15):  $S_r = \Delta R_C / \Delta D = 1000/1003 \approx 1 \Omega$ . Thus, the task is solved correctly.

The experiment was carried out on the basis of ATmega328 microcontroller of the Arduino UNO hardware platform [12]. The program is developed in Arduino IDE in C language. The used elements were:  $R = 1\text{M}\Omega$ ,  $C = 1\text{nF}$ . The time constant of the RC-filter is  $\tau = R \cdot C = 1\text{ms}$ .

## Conclusions

1. The analytical expressions presented in this paper allow calculating the main parameters of the MMC to provide the necessary resistance resolution.
2. The MMC conversion characteristic is linear in the range from 500 to 1500  $\Omega$ .
3. The MMC temperature resolution with the KTY81/110 sensor – 0.086 °C (DS18B20 – 0.0625 °C). MMC conversion time – 4 ms (DS18B20 – 750 ms).
4. The work results can be used to design intelligent sensors based on microcontrollers of other manufacturers.
5. The proposed MMC can be performed on a single semiconductor chip, which will reduce its cost.

## References

- [1] Huddleston C. Intelligent sensor design using the microchip dsPIC. Burlington, USA: Elsevier, 2007. 303 p.
- [2] Vosstrukhin A.V., Vakhtina E. A. 2015. Mikrokontrollernyj izmeritel'nyj preobrazovatel' s upravlyayemym pitaniem rezistivnykh izmeritel'nykh tsepej metodom shirotno-impul'snoj modulyatsii (Microcontroller measuring converter with controlled power supply of resistive measuring circuits by the method of pulse-width modulation), Stavropol, Russia, Pat. 2563315. Available at <http://www.findpatent.ru/patent/256/2563315.html>.
- [3] DS18B20 Programmable Resolution 1-Wire Digital Thermometer. Datasheet. [online] [11.03.2019]. Available at <https://datasheets.maximintegrated.com/en/ds/2812.pdf>.
- [4] STLM75 Datasheet (PDF) – STMicroelectronics. [online] [10.03.2019]. Available at <http://pdf1.alldatasheet.com/datasheet-pdf/view/211805/STMICROELECTRONICS/STLM75.html>.
- [5] TMP107 Digital Temperature Sensor with Bidirectional UART One-Wire Interface and EEPROM. Datasheet. [online] [08.03.2019]. Available at <http://pdf1.alldatasheet.com/datasheet-pdf/view/772883/TI/TMP107.html>.
- [6] LMT01 0.5 °C Accurate 2-Pin Digital Output Temperature Sensor With Pulse Count Interface. Datasheet. [online] [08.03.2019]. Available at <https://www.ti.com/lit/ds/symlink/lmt01.pdf>.
- [7] Vosstrukhin A.V., Vakhtina E.A. Equilibration of the Wheatstone Bridge by the Pulse-Width Modulation method // ARPN Journal of Engineering and Applied Sciences. vol. 9, No. 4, 2014, pp. 568-573. Available at [http://www.arpnjournals.com/jeas/volume\\_04\\_2014.htm](http://www.arpnjournals.com/jeas/volume_04_2014.htm).
- [8] Vosstrukhin A.V., Masteponenko M.A., Vakhtina E.A., Gabrielyan S.Z. A microcontroller measuring resistance converter. Russian Electrical Engineering, vol. 89, No. 7, 2018, pp. 421-424.
- [9] ATmega328/P. Datasheet complete. [online] [04.03.2019]. Available at [http://mkprog.ru/wp-content/uploads/2017/09/ATmega328-328P\\_Datasheet.pdf](http://mkprog.ru/wp-content/uploads/2017/09/ATmega328-328P_Datasheet.pdf).
- [10] 700 Series Platinum RTDs. [online] [06.03.2019]. Available at <https://lib.chipdip.ru/179/DOC000179754.pdf>.
- [11] KTY81 series Silicon temperature sensors. [online] [07.03.2019]. Available at [https://www.nxp.com/docs/en/data-sheet/KTY81\\_SER.pdf](https://www.nxp.com/docs/en/data-sheet/KTY81_SER.pdf).
- [12] Blum J. Exploring Arduino: Tools and Techniques for Engineering Wizardry. Wiley, 2013. 384 p.