A method of measuring RLC components for microcontroller systems

Abstract. A new method of measuring RLC components for microcontroller systems dedicated to compact smart impedance sensors based on a direct sensor-microcontroller interface is presented. In the method this direct interface composed of a reference resistor connected in series with the tested sensor impedance is stimulated by a square wave generated by the microcontroller, and then its voltage response is sampled by an internal ADC of the microcontroller. The obtained set of voltage samples is used to determine values of the sensor model impedance components.

Streszczenie. Przedstawiono nową metodę pomiaru elementów RLC dla systemów sterowanych mikrokontrolerami dedykowanymi dla inteligentnych czujników impedancji opartych na bezpośrednim interfejsie czujnik-mikrokontroler. W metodzie interfejs ten złożony z rezystora referencyjnego połączony szeregowo z badanym czujnikiem impedancji pobudzany jest przebiegiem prostokątnym generowanym przez mikrokontroler, a jego odpowiedź napięciowa próbkowana jest przez wewnętrzny przetwornik A/C mikrokontrolera. Zbiór próbek napięcia jest używany do wyznaczenia wartości elementów modelu impedancji czujnika. (Metoda pomiaru elementów RLC dla systemów sterowanych mikrokontrolerami).

Keywords: smart sensors, microcontrollers, FFT, impedance measurements.

Introduction
At present, wearable electronics and also - especially in medicine applications - swallowing electronics become more and more popular. Often, they are smart sensors (Fig. 1) which obtain information about the working environment and objects. They are placed in. They process, store and send the measurement data via any wireless interface to e.g. a smartphone, a tablet or a PC.

Smart sensors should be energy-efficient data acquisition systems, because they either are battery-powered or they take energy from their environment (energy harvesting [1,2]). Obviously, they should be small and rather cheap. This requirement can be met by the use of microcontrollers, which are widely available low-power one-chip universal devices.

Hence, a new measurement method for smart sensors based on microcontrollers and direct interfaces to impedance-based sensors modeled by two-terminal networks consisting of RLC components is proposed in the paper.

The direct interface is a voltage divider [3] consisting of a reference resistor $R_r$ working as a current-to-voltage converter and the sensor with an impedance $Z$ (Fig. 1).

The microcontroller simultaneously works as a signal generator stimulating the direct interface composed of a reference resistor $R_r$ connected in series with the sensor...
impedance (the capacitor $C$) and as a voltage measurement block sampling the voltage response $v_{out}$ of the sensor. The stimulating signal is a square wave $v_{in}$ with an amplitude set \textit{a priori} to $V_{CC} = 3.3$ V and a period $T$. The period $T$ is determined before measurements, because its value depends on the values $R_r$ and $C$ and also especially on a threshold voltage $V_{CC\text{SCLDED}} = 2.063$ V. It follows from the fact that the maximum value of $v_{out}$ should be less than a reference voltage $V_{ref}$ of the ADC ($V_{ref} = V_{CC} - 0.7$ V) \cite{4}. Owing to that, the values of $v_{out}$ are in the measurement range of the ADC.

The idea of the measurement method

Generally, the measurement method consists of the following steps:
- Determining the period $T$ of a square wave $v_{in}$ stimulating the direct interface with the tested impedance $Z$.
- The measurement procedure - sampling $M$ times the time response $v_{out}$ of the direct interface in the period $T$.
- Calculating from the set of $M$ voltage samples the real and imaginary parts of the first harmonics of the stimulating signal $v_{in}$ and the response signal $v_{out}$ basing on the FFT \cite{9}.
- Calculating the component values of the impedance $Z$ (calculating the value of capacitor $C$).

Obviously, the calculations in the two last steps depend on the impedance model. For instance, for more complex models we have to determine from the set of $M$ voltage samples additionally DC components and, if needed, the real and imaginary parts of the second harmonics.

The timing of the measurement method for the two first steps is shown in Fig. 3. It is divided into four stages.

![Fig. 3. The timing of the measurement method, $R_r = 10$ k$\Omega$, $C = 100$ nF](image)

In the first stage the \textit{determine_period} function is executed. Determination of the period $T$ takes from one pulse for $T/2 < 2.048$ ms to seven pulses for $T/2 > 2.097152$ s. It follows from the fact that the 16-bit TC0 is clocked by the 16 MHz clkPER clock \cite{4} via a prescaler, which has seven levels of the clock signal divisions: form DIV1 to DIV1024.

The sampled response $v_{out}$ should be a periodic signal. It was tested that already the fourth period of $v_{out}$ meets this condition. Hence, the \textit{determine_period} function is called for the second time (the second stage) to ensure the minimum of four periods and that - for $T/2 - V_{CC\text{SCLDED}}$ does not significantly exceed the value of $V_{CC\text{SCLDED}}$. Next, the \textit{measurement} function responsible for sampling the time response $v_{out}$ $M$ times during the period $T$ is called two times (stages three and four). The current measurement results are obtained from the second call of the \textit{measurement} function. This solution simplifies the program code. The stimulating signal should be as short as possible for power saving purposes. Therefore, the stimulating signal is ended after the fourth pulse.

Determination of the period $T$

A determination algorithm of the period $T$ of the stimulating square wave $v_{in}$ is implemented in the \textit{determine_period} function (Fig. 4).

![Fig. 4. A flowchart of the algorithm of \textit{determine_period} function](image)
the TC0 is started in the Normal Mode with the prescaler set to DIV1, the pulse is generated at the PC0 output of microcontroller, the variable end_wave is set and the variable overflow_wave is cleared. These variables are used to synchronize executions of the main code and the interrupt services.

The further execution of the algorithm depends on which interrupt occurs the first. If it is an interrupt from the AC (ACA AC0 interrupt on a rising output edge), the pulse is ended, times $t_2 = T/2$ and $t_M = T/M$ (where $M = 2^{N+1}$). Owing to this $t_2$ is an integer multiple of $t_M$, what enables to accurately determine these times in pulses by the TC0.

However, if the interrupt source is Compare on the channel C of TC0 (the TC0 Compare channel C interrupt), what means that $v_{out}$ has not reached the value $V_{SCALED}$, the pulse is generated again but with the next value DIVx of the clock signal divisions set in the prescaler ($x = 1, 2, 4, 8, 64, 256, 1024$). When this situation takes place for the division DIV1024, the maximum value $t_{MAX}$ is written to $t_2$ and its corresponding value - to $t_M$.

The TC0 Compare channel D interrupt service is used only for stopping the TC0 and for clearing the variable end_wave, what means that a given period $T$ of the stimulating square wave ended.

At the end of the function the AC is disabled, because the ADC1 input will be used by the ADC in the measurement function.

The measurement procedure

The timing of the measurement procedure implemented in the measurement function is shown in Fig. 5. For clarity, $M = 32$. In fact, in the program $M = 256$. It is the minimum value of the number of samples which enables to correctly determine the real and imaginary parts of the first harmonic of the response signal $v_{out}$.

It should be emphasized that the complete algorithm of the measurement procedure is partly contained in the software (Fig. 6) and partly “built-in” in the configuration of peripheral devices of the microcontroller (Fig. 7). Therefore, the description of the measurement procedure takes into consideration simultaneous analysis of these figures.

The code placed in the main body of the measurement function is responsible for starting the measurement procedure. The TCC0_CNT register of the TC0 is cleared, the correction value $t_{correct}$ resulting from the software delay is written to the 16-bit Compare or Capture on the channel A register TCC0_CCA, whereas to the TCC0_CCB register there is written an initial value, i.e. 1 pulse. Also, the counter of samples $i$ is cleared. This variable is also used for synchronization of the measurement function executed in the main loop of the program with the hardware. Then, the TC0 is run in the Normal Mode with the prescaler set to a determined value of clock signal divisions DIVx. Next, the measurement function is waiting for measurement of $M$ voltage samples of $v_{out}$ by the ADC ($i < M$). The obtained set of voltage samples $\{v_i\}_{i=1}^{M}$ are stored in a table of 16-bit variables $v_{sample[M]}$.

---

**Fig. 6. A flowchart of the algorithm of measurement procedure**

**Fig. 7. A block scheme of the sensor part configured from peripheral devices of the ATXmega32A4 microcontroller**

Counting 1 pulse by the TC0 generates the TC0 Compare channel B interrupt, which starts serving the stimulating pulse in the time $t_1$ — the high level at the PC0 output of the microcontroller is changed to the low level (the stimulating signal passes the inverting buffer - Fig. 2). Next, the value written in the TCC0_CNT register is increased by adding the value $t_2$.

The next call of this interrupt ends generation of the stimulating pulse. The low level of the signal $v_{in}$ lasts $t_2$.

The events of measuring Compare between the TCC0_CNT register and successive values of the TCC0_CCA register, i.e. the ends of countdown of the lines $t_i = t_0 + t_{corr}$, $i = 0, ..., M - 1$, trigger - via the event system - measurements of the voltage $v_{out}$ at the ADC1 input on the channel 0 of the ADC (Fig. 7). This solution eliminates software delays and simplifies the algorithm of measurement procedure.

These events generate also interrupts (Fig. 7) (the TC0 Compare channel A interrupt), during their service the time $t_i$ is updated - the value $t_{id}$ is added to the value stored in the TCC0_CCA register.

When the ADC ends the next measurement of voltage $v_{out}$, the ADC A channel 0 Conversion Complete interrupt is generated. During its service the 12-bit voltage measurement result stored in the 16-bit ADCA_CH0RES register is saved to the $i$th element of the v_sample array and the counter of samples $i$ is incremented.

When the counter $i$ reaches the value $M$ the loop of waiting for these results is ended, the TC0 is stopped and generation of the stimulating square wave is ended.

### Calculation of the impedance component values

The real $v_{re\_out}$ and imaginary $v_{im\_out}$ parts of the first harmonic of the response signal $v_{out}$ are calculated from the set \( \{v_{re\_out}, v_{im\_out}\} \) by a custom function based on the FFT. A shape of $v_{in}$ is known, hence the values $v_{re\_in}$ and $v_{im\_in}$ of its first harmonic are constants written in the program code, what simplifies calculations and reduces the code size.

Therefore, the capacitor $C$ value is calculated from the following formula (1):

\[
C = \frac{1 - K_1}{K_1 \cdot \omega^2 \cdot R_C^2},
\]

where:

\[
\omega = 4 \cdot \pi \cdot t_2 \cdot DV_{ref} \cdot t_{CLK}, \quad t_{CLK} = 0.0625 \mu s
\]

\[
K_1 = \frac{v_{re\_out}^2 + v_{im\_out}^2}{v_{re\_in}^2 + v_{im\_in}^2}
\]

### Conclusions

In the paper a new compact smart impedance sensor solution based on a direct sensor-microcontroller interface and a new method of measuring RLC components for microcontroller systems with internal ADCs is presented. In the method the direct interface consisting of a resistor $R_C$ connected in series with the tested sensor impedance is stimulated by a square wave generated by the microcontroller. Then, its voltage response is sampled by an internal ADC of the microcontroller. The obtained set of voltage samples is used to determine values of the impedance components.

An example of complete application of a compact smart impedance sensor for the ATXmega32A4 and details of determining the duration of the stimulating signal and the measurement procedure algorithms are also presented.

Summarizing, the proposed approach enables to obtain a low-cost and low-power solution of a smart sensor. It follows from the fact that the proposed simple measurement procedure needs a short measurement time, and determination of the values of impedance model components does not require complicated calculations. Additionally, a small usage of program and data memories by software is needed (codes of the `determine_period` and `measurement` functions occupy 732 B, and codes of calculation functions take 3058 B — the code was written in the C language and compiled by the AVR GCC in the Atmel Studio 6).

These advantages, especially a low cost and a low power consumption, give possibilities of using the compact smart sensors in both battery-powered and wireless sensor networks.

**Autor:** dr hab. inż. Zbigniew Czaja, Gdańsk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Department of Metrology and Optoelectronics, ul. G. Narutowicza 11/12, 80-233 Gdańsk, E-mail: zbczaja@pg.gda.pl.

**REFERENCES**


