Post-print of: Czaja Z.: A measurement method for capacitive sensors based on a versatile direct sensor-to-microcontroller interface circuit MEASUREMENT Vol. 155 (2020), pp.1-11 DOI:10.1016/j.measurement.2020.107547

A measurement method for capacitive sensors based on a versatile direct sensor-to-microcontroller interface circuit

Zbigniew Czaja

Gdansk University of Technology,
Faculty of Electronics, Telecommunications and Informatics,
Department of Metrology and Optoelectronics,
ul. G. Narutowicza 11/12, 80-233 Gdansk, Poland,
email: zbczaja@pg.edu.pl.

Abstract

In the paper, there is presented a new time-domain measurement method for determining the capacitance values of capacitive sensors, dedicated, among others, to capacitive relative humidity sensors. The method is based on a versatile direct sensor-to-microcontroller interface for microcontrollers with internal analog comparators (ACs) and with precision voltage reference sources, e.g. digital-to-analog converters (DACs). The reference source can be replaced by a resistive divider attached to the negative input of the AC. The interface circuit consists only of a reference resistor R_r, a given capacitive sensor working as a voltage divider, and a microcontroller (its peripherals: AC, timer, DAC, I/O pins). A prototype of the proposed complete solution of a compact capacitive smart sensor based on an 8-bit ATXmega32A4 microcontroller has been developed and tested. The maximum possible relative inaccuracy of an indirectly measurable capacitance was analysed, and experimental research was also performed. The results confirmed that the relative errors of value determination for a capacitive sensor are less than ±0.06%, which corresponds to a capacitance measurement accuracy of less than 0.1 pF for a range of measured capacity values from 100 pF to 225 pF, which in turn corresponds to at least a 0.5% relative humidity resolution for commercial capacitive RH sensors (e.g. TE Connectivity HS1101LF and Philips H1).

Keywords: microcontroller interfacing, humidity sensors, ADCs, time-domain measurement

1. Introduction

Humidity sensors are an important group of capacitive sensors. They are used wherever humidity affects a production process (in technological process control and diagnostics quality

control, e.g. in manufacturing of paper, textiles, tobacco, food and even integrated circuits), comfort and safety of life (e.g. in archive rooms, libraries, museums, offices, and increasingly at homes, for example in air conditioning equipment), or scientific research [1]. At present, low-cost—that is, miniaturized—humidity sensors are produced using such technologies as capacitive, hygrometric, gravimetric, optical and integrated (as additional on-chip transducing components) ones [2]. However, it should be emphasized that low-cost hygrometers usually use capacitive relative humidity (RH) sensors because they are characterized by low energy consumption, are non-intrusive and non-invasive, emit no radiation and have short response times [2-5].

In capacitive sensors, the changes of capacitance values, more precisely dielectric changes of thin films upon water vapor uptake, depend on the ambient RH changes. As presented in [6] for commercial capacitive RH sensors [7,8], the sensitivity varies from 0.2 pF up to 0.5 pF per 1% of RH, and the capacitance is between 100 pF and 500 pF at about 50% of RH at 25°C.

Currently, RH sensors are often designed as smart sensors. They consist of capacitive RH sensors and additional blocks: conditioning circuits, processing and control units, and communication interfaces. Thanks to this, they can either be built into devices as parts of control systems or work as independent devices. Therefore, apart from measuring RH, smart sensors process and store the measurement results. They can send the results either to the main controller of the monitored device or - via a wireless interface - to a smartphone or a personal computer. Also, they can function as parts of wireless sensor networks.

In the second case, the smart sensors are either battery-powered or gain their power from the environment (energy harvesting [9]). Therefore, the smart sensors working as data acquisition systems should be energy-efficient, and they should be as small and simple as possible to reduce cost [10]. For this reason, the use of microcontrollers can be the best solution, because they are low-power, one-chip universal devices simultaneously performing the functions of control and measurement systems, data processing and storage, acting also as communication interface controllers.

For further cost reduction of a smart sensor, we can reduce the cost of the interface circuits by employing direct sensor-to-microcontroller interfaces, which are used for capacitive sensors [6,11-18] and also for inductive sensors [19,20]. In the first case, as described in [6], the interface circuit consists of a capacitive sensor C_x, a calibration capacitor C_c and two resistors R_i and R_d (a stray capacitance C_{off} between a pin of the microcontroller and ground is also taken into account). Hence, the determination of the sensor capacitance C_x value is based on the internal timer of the microcontroller measuring the discharging times of $R_d(C_x + C_{off})$, $R_d(C_c + C_{off})$ and R_dC_{off} circuits. However, it is still possible to simplify the interface circuit, as described in [21-23]. The proposed versatile and simple circuit consists only of a reference resistor R_r and a capacitive sensor C_x [23].

This has become possible through the use of internal analog devices present in almost all modern 8-bit and especially 32-bit microcontrollers: analog-to-digital converters (ADCs) and analog comparators (ACs).

Thus, in this paper there is proposed and analyzed a new time-domain measurement method for capacitive sensors, determining capacitance values of the capacitive RH sensors based on a versatile direct sensor-to-microcontroller interface circuit [23]. In this method one electrode of the capacitive sensor is connected to GND of the interface circuit (Fig. 1). Therefore, the method is ~~even~~ suitable for grounded capacitive sensors [24,25]. The method is a modification of the method proposed in [23] including, among others:

- replacing ADC with a precise voltage source, which can reduce energy consumption (it depends on the used reference source),
- eliminating the logarithm calculations performed by the microcontroller during the measurement procedure, which significantly shortens its execution time and thus reduces energy consumption,
- adjusting the interface circuit to measure capacitances of tens and hundreds of pF.
- comprehensive extending the measurement procedure to a series of multiple measurements what enables to significantly improve the measurement capacitance accuracy resulting from a significant reduction of measurement errors.

It should be emphasized that the hardware configuration and the way of stimulation of the interface circuit do not change, which enhances versatility of this interface circuit. Therefore, to adapt the system to measuring small capacitances and to reduce energy consumption, it is enough to replace the microcontroller software by the software that implements the new method, which is an advantage from a practical point of view.

To analyze and experimentally verify the proposed method, the capacitive RH sensors are replaced by capacitors with capacitance values from a typical range specified in [6], where two commercial sensors [7,8] are accurately characterized.

The paper is organized as follows:

- Section 2 presents the operating principle of the measurement method,
- Section 3 analyses the error sources of the capacitance measurements and their limitations,
- Section 4 describes the experimental results and proposals to improve measuring accuracy,
- Section 5 compares the method with the state of the art,
- Section 6 contains the main conclusions.

2. Operating principle

2.1. The operating principle of previous methods

In Fig. 1, we can see a direct versatile sensor-to-microcontroller interface circuit, which works according to the time-domain measurement method described in [22,23]. This circuit, consisting of an impedance Z representing the sensor (in our case a capacitive sensor) and a reference resistor R_r working as a current-to-voltage converter, acts as a voltage divider. A square pulse v_{in} with an amplitude set *a priori* to the supply voltage V_{CC} of the microcontroller generated at its Pin1 stimulates this divider. The voltage response signal v_{out} of the circuit is obtained at Pin2. The inputs of two internal measurement peripherals of the microcontroller (an analog comparator (AC) and an analog-to-digital converter (ADC)) are connected to this pin. The AC is used to trigger the ADC and the timer via the internal connection system of the microcontroller, when the voltage response v_{out} reaches a fixed voltage level $V_{threshold}$. Hence, the ADC samples signal v_{out} and the timer specifies moment t_m of sample V_m relative to the beginning of signal v_{in} [23]. Theoretically, V_m should be equal to $V_{threshold}$, however, typical voltage reference sources built into microcontrollers are inaccurate and usually not adjustable. For this reason, the AC only determines the voltage level at which the ADC performs accurate voltage measurements.

2.2. The new operating principle

The proposed solution is characterized by the fact that we only need to use a precision and tunable voltage reference source to eliminate the use of ADC, which reduces energy consumption (described in Subsection 2.6), primarily by eliminating the logarithm calculations performed during the measurement procedure. In this case, we can assume that $V_m = V_{threshold}$. Thanks to this, there is no need for the microcontroller to calculate the logarithm function to determine capacitance values of the capacitive sensor based on the measured values of V_m [23], which shortens the measurement procedure. It can be done only once after determining the threshold value V_m for a given sensor – during the design of the interface circuit. Thus, this value is treated as a constant in the calculations of the capacitance values by the measurement procedure. In the paper, it is proposed to use a digital-to-analog converter (DAC) implemented in the microcontroller as the reference source, as shown in Fig. 2. It meets all of the above requirements. DACs are as accurate as ADCs, and the voltage values at their output are set by software. In addition, a 12-bit DAC is about 35 times more accurate than a 6-bit voltage scaler in the AC (taking into account 6 LSB inaccuracy of DAC) [26]. It should be also mentioned that the DAC consumes less power (less than 1.1 mA in the low-power mode – when the DAC works as the voltage reference for the AC its output driver and output pin of channel 0 are disabled) than the ADC (2.9 mA) [27, page 62].

It should be mentioned that in the target applications instead of a DAC we can use a resistive divider consisting of R_1 and R_2 (as shown in Fig. 1), what inter alia enables to use microcontrollers without DACs. It is possible, because the threshold value V_m is constant for a given sensor, as presented earlier.

In Fig. 2, an example of a versatile direct sensor-to-microcontroller interface circuit for capacitive sensors based on an 8-bit ATXmega32A4 microcontroller [24] is shown. This microcontroller was chosen because, as mentioned in [23], any application which can run on 8-bit microcontrollers can also run on 32-bit microcontrollers, which makes the proposed solution more versatile. Additionally, this microcontroller has all necessary peripherals to implement the proposed method: a 12-bit DAC, an AC, a 16-bit timer/counter and a 12-bit ADC.

The microcontroller performs two functions during the measurement procedure. The first one generates stimulation signal v_{in} at output pin PC0, and the second - measures time t_m by timer TC0 for which signal v_{out} at pin PA1 reaches value V_m set by the DAC (the AC sends a trigger signal via the event system to the timer working in the input-capture mode), as shown in Fig. 3. After this time the generation of signal v_{in} is stopped, reducing energy consumption by the interface circuit.

2.3. The algorithm of the measurement procedure

The details of the measurement procedure are included in the flowchart of its algorithm presented in Fig. 4. The algorithm is implemented in two code sections: the main function section and the interrupt services section, and in the configuration of peripheral devices of the microcontroller forming the measurement microsystem (the time measurement block in Fig. 2). The code placed in the main body of the measurement function starts timer TC0 in Normal Mode and starts the generation of signal v_{in} (the high level is set at pin PC0). It then configures and runs the time measurement block. Also, flag *end_conv* used to synchronize the software and hardware, is set. Next, we wait for completion of the measurement of time t_m by testing flag *end_conv*. When the AC detects the voltage v_{out} value achieves the value of voltage V_m, it generates an event, which saves the current value of the timer to the capture register of channel A. In turn, this event generates an interrupt. As a result, the measured time t_m is saved and flag *end_conv* is cleared - causing the transition to the main function code. After that, signal v_{in} is stopped and the time measurement block is turned off.

2.4. Circuit analysis

As shown in Fig. 2, a divider's voltage ratio $K(s)$ of the versatile direct sensor-to-microcontroller interface circuit is as follows (1):

$$K(s) = \frac{v_{out}(s)}{v_{in}(s)} = \frac{Z(s)}{Z(s) + R_r} \tag{1}$$

Hence, the formula for v_{out} has the form (2):

$$v_{out}(s) = v_{in}(s) \cdot \frac{Z(s)}{Z(s) + R_r} \tag{2}$$

Because all of the measurements are carried out only during pulse duration T, $t_m < T$, the stimulating signal v_{in} of the sensor interface circuit in the time domain can be written as:

$$v_{in}(t) = V_{in} \cdot (\mathbf{1}(t)) \quad \text{only for} \quad t \in (0; T) \tag{3}$$

and therefore, taking into account this assumption, in the frequency domain we can use for our analysis its simplified form:

$$v_{in}(s) = V_{in} \cdot \frac{1}{s}, \tag{4}$$

where $V_{in} = V_{CC}$ is a pulse amplitude.

A simplified model of the interface circuit, taking into account the pin parameters of the microcontroller, is shown in Fig. 5. E.g. for an ATXmega32A4 microcontroller, pin capacitance C_{pin} is around 10 pF, output pin resistance R_{pin_on} at a high voltage level is around 50 Ω, and input pin resistance R_{pin_off} is estimated at around 3 GΩ [27, page 67]. We assumed, as presented in [6], that the values of capacitive sensor capacitance C_x are in a range between 100 pF and 220 pF. Therefore, these pin parameters have a significant share in the circuit model. We should also take into account parasitic capacitances C_{p1} and C_{p2} representing inter alia capacitances between printed circuit board traces. If ADC is used, we have to add to the model a serial connection of C_{sample} (5 pF) and $R_{source+channel}$ (4.5 kΩ) [28].

The circuit model shown in Fig. 5 can be simplified to the model presented in Fig. 6. In this case, $C_1 = C_{pin} + C_{p1}$ and $C_2 = C_x + (C_{p2} + C_{pin})$. Based on this model, and taking into account the fact that $R_{pin\_off} >> R_r$ (R_r is equal to about 10 MΩ, as described in the next subsection), the formula for $K(s)$ of the capacitive sensor interface circuit was derived, and it has the form:

$$K(s) = \frac{1}{s(R_{pin\_on} \cdot C_1) + 1} \cdot \frac{1}{s(R_r \cdot C_2) + 1} \tag{5}$$

Because $R_{pin\_on} \cdot C_1 << R_r \cdot C_2$ (values of C_1 and C_2 are at a similar level, but R_{pin_on} is six orders smaller than R_r), we can simplify the formula (5) to the form:

$$K(s) = \frac{1}{s(R_r \cdot C_2)+1} \tag{6}$$

Based on formulae (2), (4) and (6), we can derive the formulae for voltage response v_{out} of the interface circuit in the frequency domain:

$$v_{out}(s) = V_{in} \cdot \left(\frac{1}{s} \cdot \frac{1}{s(R_r C_2)+1} \right) \tag{7}$$

and in the time domain:

$$v_{out}(t) = V_{in} \cdot \left(1 - e^{\frac{-t}{R_r C_2}} \right) \text{ for } t \in (0;T) \tag{8}$$

From (8), we can determine the formula for C_2:

$$C_2 = \frac{-t_m}{R_r \cdot \ln\left(1 - \frac{V_m}{V_{in}}\right)} \tag{9}$$

Because $C_2 = C_x + (C_{p2} + C_{pin})$, it enables to determine the capacitive sensor capacitance C_x:

$$C_x = \frac{-t_m}{R_r \cdot \ln\left(1 - \frac{V_m}{V_{in}}\right)} - (C_{p2} + C_{pin}) \tag{10}$$

The values of C_x depend on the measurement result t_m, the settings of the interface circuit (more precisely, the setting of R_r resistor value) and the voltage reference source (the voltage ratio V_m/V_{in}) and also on the constant parameters of the electronic system ($C_{p2} + C_{pin}$).

Because $R_{pin\_on} << R_r$, we can assume that $V_{in} = V_{cc}$. The voltage reference of the DAC, working as the voltage reference source, is set to V_{cc}, hence we can write that $V_m = \alpha \cdot V_{cc}$, where $\alpha = code_{DAC}/2^n$, where $code_{DAC}$ is the code written to the DAC data register, and n is the DAC resolution. We can also assume that time t_m, measured by the timer/counter of the microcontroller is given: $t_m = m_t \cdot t_{clk}$, where t_{clk} is a period of the peripheral clock (synchronous with the CPU clock [26]) which clocks the timer, and m_t is the measurement result of time t_m stored in the timer data register.

Therefore, formula (10) can be written in the following form:

$$C_x = m_t \cdot \frac{-t_{clk}}{R_r \cdot \ln(1-\alpha)} - (C_{p2} + C_{pin}) \tag{11}$$

Thanks to this V_{cc} does not depend on the determined values of C_x. It is enough for V_{cc} to be constant throughout the entire measurement cycle. Thus, the values of C_x depend on α and t_{clk} which are constant for a given microcontroller.

If we use a resistive divider as the reference voltage (Fig. 1), $\alpha = R_2 / (R_1 + R_2)$ is also constant and independent of V_{cc} for $R_1 + R_2 >> R_{pin\_on}$, e.g. for $\alpha = 0.5378$, $R_1 = 232$ kΩ, $R_2 = 270$ kΩ.

If we assume that $C_c = C_{p2} + C_{pin}$ and $\beta = -t_{clk} /(R_r \cdot \ln(1 - \alpha))$, where $0 < \alpha < 1$, we obtain the final formula for C_x:

$$C_x = m_t \cdot \beta - C_c \tag{12}$$

In this case, parameter β characterizes the configuration of the interface circuit and can be calculated only once during the design of the electronic system. In turn, parameter C_c represents the physical properties of the interface circuit, and it can be either determined by the PCB design software or measured. Both parameters are constant for a given interface configuration, and they can be saved as constants in the program of the microcontroller. Thanks to this, calculations of C_x based on (12) are very quick and simple in contrast to the method presented in [23].

2.5. Determination of the reference resistor R_r value

Measurement of time t_m should be made with the best possible resolution of the microcontroller's timer. Hence, for a given range of capacitance value changes (from C_{x_min} to C_{x_max}) of a capacitive sensor, this time value should be greater than $t_{m\_min} = m_{t\_min} \cdot t_{clk}$, but less than $t_{m\_max} = 2^N \cdot t_{clk}$, where N is the timer resolution. The value m_{t_min} determines the maximum value of a quantization error. Thus, we have to establish this value according to our requirements, and next to determine the R_r value [23]:

$$\frac{-m_{t\_min} \cdot t_{clk}}{C_{x\_min} \cdot \ln(1-\alpha)} \le R_r < \frac{-2^N \cdot t_{clk}}{C_{x\_max} \cdot \ln(1-\alpha)} \tag{13}$$

We should remember that the greater the value m_{t_min}, the greater the value R_r and the greater the value of t_m for a given value of C_x, as mentioned in [23], which improves the accuracy of measurement of time t_m by the timer [29,30], but it lengthens time t_m what unfortunately increases energy consumption. This fact is disadvantageous, especially for battery-powered systems. In conclusion, this value should be a compromise between the accuracy of measurement of time t_m and the energy consumed by the system during the measurement procedure.

For our example of interface circuit, we assume $m_{t\_min} = 2^{13}$, so that the quantization error of the timer is smaller than the quantization error of a 12-bit DAC. From (12) and for coefficient $\alpha = 0.5$

(determination of this value is described in the next section) and C_x ranging from C_{x_min} = 100 pF to C_{x_max} = 220 pF, the value R_r should be in a range of 7.39 MΩ to 26.86 MΩ. Hence, R_r = 10 MΩ was chosen. For this value, the estimated time t_m is in a range of 0.9 ms to 1.86 ms.

2.6. Power consumption analysis

The AC and the timer work throughout time t_m. The AC current consumption (low-speed) is 110 μA, and the timer current consumption (Prescaler DIV1) is 19 μA [27, page 62], what gives a total of about 0.13 mA. The maximum value of current flowing through pin PC0 and consumed by the interface circuit R_r and C_x is 0.33 μA, so it can be neglected. The DAC consumes less than 1.1 mA in the low-power mode (we do not use the output driver and the output pin of channel 0 of the DAC). The DAC can be either used throughout time t_m or switched on before the expected AC comparison. Assuming the first unfavorable option, the power consumption is 1.23 mA by the time t_m (0.9 ms to 1.86 ms) what gives the maximum value of energy consumption in between 3.65 μJ and 7.55 μJ.

For the ATXmega32A4 microcontroller the power supply current in the active mode at 3.3 V, for a 16 MHz external clock and at 25°C is equal to about 7.5 mA [27, page 70]. The calculation time of the natural log function optimized by the author is only 900 μs (18 iterations). Hence, to determine the capacitive sensor values [23], 1000 μs is needed. Thus, the energy consumption is 24.75 μJ, and it is at least 3.3 or 6.8 times bigger (depending on the measured value of C_x) than the energy consumption by the peripherals throughout time t_m. Therefore, it can be concluded that eliminating the logarithm calculations significantly reduces the energy consumed during the measurement procedure.

3. Error analysis

The maximum possible relative inaccuracy (error) $\Delta y / |y|$ of an indirectly measurable variable y [31], which is an approximation of the standard uncertainty [32], will be used to estimate the inaccuracy of the determination of C_x values of capacitive sensors from the measurement results:

$$\frac{\Delta y}{|y|} = \frac{1}{|y|} \cdot \sum_{i=1}^{I} \left| \frac{\partial f}{\partial x_i} \right| \cdot |\Delta x_i| , \tag{14}$$

where: $y = f(x_1, x_2, .. , x_I) \neq 0$ is an indirectly measurable variable, $x_1, x_2, .. , x_I$ are directly measurable variables, Δy is the maximum absolute inaccuracy of function f, and Δx_i is the maximum absolute inaccuracy of a directly measurable variable x_i (i = 1, 2, .., I).

If we assume that $C_m = m_t \cdot \beta$ (C_c is constant and independent of measurements), based on (12) and (14), the maximum possible relative inaccuracy $\Delta C_x / |C_x|$ taking into account an indirectly measurable capacitance $C_m = C(m_t, \beta)$ has the following form:

$$\frac{\Delta C_x}{C_x} = \frac{\Delta C_m}{C_m} + \frac{\Delta C_c}{C_c} = \frac{\Delta \beta}{\beta} + \frac{\Delta m_t}{m_t} + \frac{\Delta C_c}{C_c}, \tag{15}$$

where: $C_m > 0$, $\beta > 0$, $m_t > 0$ and $\Delta C_c/C_c$ is also constant and independent of measurements.

From (11), it can be seen that β is a function of t_{clk}, R_r and α. Hence, we can write:

$$\frac{\Delta \beta}{\beta} = \frac{\Delta R_r}{R_r} + \frac{\Delta t_{clk}}{t_{clk}} + \delta\alpha, \tag{16}$$

where:

$$\delta\alpha = \frac{\Delta\alpha}{(1-\alpha)\cdot \ln(1-\alpha)} \tag{17}$$

The first component $\Delta R_r / R_r$ of formula (16) can be interpreted as the tolerance of R_r [23]. E.g. if we assume a 0.01% tolerance of R_r or we measure the value of R_r with a precise multi-meter (e.g. an Agilent 34410A measures resistance with accuracy of 0.0065% [33]), the value $\Delta R_r / R_r$ is small in comparison with the third component of formula (16). For the same reason, also the influence of the thermal drift affecting R_r can be neglected, because this resistor has the temperature coefficient of resistance equal to 50 ppm/^{o}C, which corresponds to changes in the resistance value of only 0.005% per 1 ^{o}C.

The second component $\Delta t_{clk} / t_{clk}$ depends on the stability of the timer [29,30], that is, the stability of the crystal oscillator connected to the microcontroller.

The third component $\delta\alpha$ (17) represents the inaccuracy of the DAC conversion, and generally it depends on the quality of performance of the DAC [27, page 66]. A graph of $\delta\alpha$ for $\Delta\alpha = 6$ LSB is shown in Fig. 7. It can be seen that $\delta\alpha$ values are minimal (less than 0.5%) for α between 0.5 and 0.75. Hence, $\alpha = 0.5378$ was chosen. It should be mentioned that $\Delta\alpha$, so also $\delta\alpha$, is constant for a given α value and a given microcontroller, and it was not observed that α was sensitive to temperature changes, which probably results from a proper construction of the DAC, thanks to which the temperature compensation takes place.

In the case of a resistive divider the value $\delta\alpha$ can be interpreted as the sum of tolerances of resistors R_1 and R_2. It should be emphasized that if we use resistors with the same temperature coefficients of resistance, α also will not be sensitive to temperature changes.

In turn, the second component $\Delta m_t / m_t$ in (15) can be written:

$$\frac{\Delta m_t}{m_t} = \frac{1}{m_t} + \frac{\Delta m_{delay}}{m_t} + \frac{\Delta m_{trigger}}{m_t} \tag{18}$$

The first component in (18) is the quantization error. For the set values of interface circuit components, this error is less than 0.009%.

The second component consists of $\Delta m_{delay} = \Delta m_{delay\_AC} + \Delta m_{delay\_Event\_System}$ and is a constant propagation delay introduced by the AC and the internal connection system of microcontroller (the event system) [26,27]. It is equal to $4 \cdot t_{clk}$. Its correction can be included in the calculation.

Whereas the last component represents the inaccuracy of the AC work, the value $\Delta m_{trigger} = f(v_{trigger})$ results from the trigger instability of AC, i.e. the instability of AC trigger voltage $v_{trigger}$, where $v_{trigger} = v_{off} + v_{noise}$, v_{off} is an input offset voltage ($v_{off} \leq 10$ mV [27, page 66] and is constant for a given value of input voltage level and a given microcontroller) which can enable to minimize the impact of this voltage on the measurement accuracy, and v_{noise} is the noise (disturbances) generated, inter alia, by the digital circuits of microcontroller (e.g. by its core processor, timers and clock system).

To determine the value $\delta m_{trigger} = \Delta m_{trigger} / m_t$, we can transform formula (11) to the following form:

$$t_m = m_t \cdot t_{clk} = -C_m \cdot R_r \cdot \ln\left(1 - \alpha_{trigger}\right), \tag{19}$$

where

$$\alpha_{trigger} = \frac{\alpha \cdot V_{cc} \pm v_{trigger}}{V_{cc}} = \alpha \pm \frac{v_{trigger}}{V_{cc}} \tag{20}$$

Substituting (20) to (19) and making some transformations we obtain the following :

$$\delta m_{trigger} = \frac{\Delta m_{trigger}}{m_t} = \frac{1}{\ln\left(1-\alpha\right)} \cdot \ln\left(\frac{1 - \alpha - \dfrac{v_{off} + v_{noise}}{V_{cc}}}{1 - \alpha + \dfrac{v_{off} + v_{noise}}{V_{cc}}} \right) \tag{21}$$

A graph of $\delta m_{trigger}$ as a function of v_{noise} is drawn in Fig. 8. For clarity of argument it was assumed that v_{off} is equal to 0 V, because it is constant for given measurement conditions, as it was previously described. It can be seen that an increase in the noise level v_{noise} of 1 mV results in an increase of the error by about 0.2%. Hence, the level of noise v_{noise} at the input of AC has the greatest impact on the accuracy of measurement of t_m, and thus on the error of determining the value of C_x.

4. Experimental results and discussion

The main aim of the experiments was to examine the metrological possibilities of a versatile direct sensor-to-microcontroller interface circuit (Fig. 2) working according to the proposed time-domain measurement method. A prototype of the complete solution of a compact smart capacitive sensor based on an 8-bit ATXmega32A4 microcontroller was used. It was not placed in any shielding box. The ATXmega32A4 microcontroller runs on a 16 MHz crystal oscillator. Its timer/counter TC0, used to measure time t_m, is clocked directly by the system clock, so t_{clk} = 62.5 ns. The value written to the DAC data register DACB_CH0DATA is 2203 (α = 0.537842). The supply voltage value measured by the Agilent 34410A Digital Multimeter is V_{CC} = 3.3000 V, and the measured value of the reference resistor R_r = 10.0654 MΩ.

The capacitive sensor was modelled by a set of nine ceramic capacitors with the following reference values of C_x measured by the Agilent 4263B LCR METER at 1 kHz by using the 16047A Test Fixture of Hewlett-Packard: 100.09 pF, 115.10 pF, 130.01 pF, 144.63 pF, 163.00 pF, 178.01 pF, 192.92 pF, 207.54 pF, 225.11 pF.

The measurements were carried out 128 times for each of the nine $C_{x,i}$ values (i = 1, .., 9). After each measurement, the measured value of time $m_{t,i,j}$ and that calculated based on (9) by the microcontroller capacitance values $C_{m,i,j}$ were sent to a personal computer via a USB interface.

Each time, for each i-th value $C_{x,i}$ of the reference capacitance, the lowest $m_{t\_min,i} = \min_j \{m_{t,i,j}\}$, $C_{m\_min,i} = \min_j \{C_{m,i,j}\}$, highest $m_{t\_max,i} = \max_j \{m_{t,i,j}\}$, $C_{m\_max,i} = \max_j \{C_{m,i,j}\}$ and middle $m_{t\_mean,i} = (m_{t\_min,i} + m_{t\_max,i})/2$, $C_{m\_mean,i} = (C_{m\_min,i} + C_{m\_max,i})/2$ values from the sets $\{m_{t,i,j}\}_{j = 1, .., 128}$ and $\{C_{m,i,j}\}_{j = 1, .., 128}$ of measurement results were determined and calculated. The lowest and highest values are plotted in the form of points, whereas the middle values are used to plot the curves presented in Fig. 9, where the results of scaling of the interface circuit for time t_m values are given in Fig. 9a for capacitances C_m directly determined from t_m in Fig. 9b. It can be seen in Fig. 9b that the scaling curve is not as ideal as we could expect ($C_m = C_2$). It is burdened with a big offset error (resulting inter alia from C_c) and also with a gain error (imperfect measurement part of the interface circuit – the β parameter). Hence, based on the Matlab *polyfit* function, the aggregated measurement results $\{m_{t\_mean,i}\}_{i = 1, .., 9}$ and the set of reference capacitance values $\{C_{x,i}\}_{i = 1, .., 9}$ were used to determine parameters β_{cort} and C_{c_corr} describing the actual configuration of the interface circuit. Therefore, the capacitance values are finally determined by the microcontroller based on the following updated formula (12):

$$C_{m\_corr,i,j} = m_{t,i,j} \cdot \beta_{corr} - C_{c\_corr} \tag{22}$$

Thus, the actual scaling curve describing the tested interface circuit is drawn in Fig. 9c. For this circuit β_{cort} = 0.008173771653 pF and C_{c_corr} = 17.43 pF.

The relative errors δC_m of capacitance determination based on (22) are drawn in Fig. 10a. The vertical lines join points representing the maximum $\delta C_{m\_max,i}$ and minimum $\delta C_{m\_min,i}$ values of relative errors for a given $C_{x,i}$, where:

$$\delta C_{m\_\max,i} = \frac{\max\limits_{j=1,..,128}\{C_{m\_corr,i,j}\} - C_{x,i}}{C_{x,i}} \cdot 100\% \text{ and } \delta C_{m\_\min,i} = \frac{\min\limits_{j=1,..,128}\{C_{m\_corr,i,j}\} - C_{x,i}}{C_{x,i}} \cdot 100\% \tag{23}$$

Analyzing Fig. 10a, we can observe that the corrected capacitance determination (22) has no effect on errors due to disturbances at the input pin of AC. They are at a high level of about 1%. Such big errors cause a low capacitance measurement accuracy ΔC_m of C_m that varies from 1.2 pF up to 2.1 pF, as shown in Fig. 10b, where:

$$\Delta C_{m,i} = \max_{j=1,..,128}\{C_{m\_corr,i,j}\} - \min_{j=1,..,128}\{C_{m\_corr,i,j}\} \tag{24}$$

As was described in [6], this accuracy should be less than 0.2 pF for the full capacitance range of a capacitive sensor to obtain an RH resolution at a level of 1%. For this reason, single measurements of capacitance are useless.

Hence, a new approach is proposed based on a series of multiple measurements of time $m_{t,m}$ (m = 1, .., M), determination of the average value m_{t_avr} from these results $\{m_{t,m}\}_{m = 1, .., M}$ and next calculation of the capacitance value from (22).

4.1. The method of multiple measurements of time m_t

A flowchart of the measurement procedure for multiple measurements of time $m_{t,m}$ (the M-procedure) is shown in Fig. 11. Generally speaking, the algorithm consists of four parts, where the first two parts are repeated M times:

- the measurement part performing a single measurement of time $m_{t,m}$ according to the algorithm shown in Fig. 4,
- the discharge procedure,
- handling the main loop,
- calculation of the average value m_{t_avr} from M measurements.

During the discharge procedure, the DAC and AC are turned off to reduce energy consumption, and pins PA1 and PC0, to which the interface circuit is connected, are set as output pins with a low voltage level to quickly discharge the capacitances of the interface circuit (Fig. 12 a). Hence, capacitor $C_1 = C_{p1} + C_{pin}$ is discharged via resistance R_{pin_on} of pin PC0, and capacitor $C_2 = C_x + (C_{p2} + C_{pin})$ via resistance R_{pin_on} of pin PA1 (R_{pin_on} is about 27 Ω [27, page 82]), as shown in a simplified schematic diagram ($R_{pin\_on} << R_r$) in Fig. 12 b. It is experimentally verified that it is sufficient for the discharging time to be 16 times smaller ($\xi = 1/16$) than m_{t_m}. Hence, the duration t_M of the M-procedure is:

$$t_M = t_{clk} \cdot \sum_{m=1}^{M} \left(m_{t,m} + \xi \cdot m_{t,m} \right) + t_{delay} \cdot M + t_{delay\_calc}, \tag{25}$$

where: t_{delay}, t_{delay_calc} are software delays.

E.g. for the assumed capacitance range of a capacitive sensor, $t_{clk} = 0.0625$ μs and $M = 64$, the value t_M is in a range between about 61.2 ms and 126.5 ms.

To determine an appropriate M value for the application, the M-procedure was run for capacitance $C_{x,1}$ and the following values $M = 2^n$, where $n = 4,5,6,7,8,9,10$. A multiplicity of 2 was chosen, inter alia, to simplify the calculation of m_{t_avr}. The measurement results are shown in Fig. 13. It can be seen that a capacitance measurement accuracy ΔC_m is approximately inversely proportional to the number n ($M = 2^n$) up to $M = 128$, then it slowly decreases to fall close to a level of about 0.03 pF.

The choice of value M should be a compromise between the capacitance measurement accuracy, i.e. the measurement resolution, and the measurement time. Therefore, $M = 64$ was chosen. In this case, the accuracy is equal to about 0.07 pF, that is much smaller than 0.2 pF [6], and the duration of the measurement procedure is acceptable.

Also in this case, measurements based on the M-procedure were carried out 128 times for each of the nine $C_{x,i}$ values. The measured value of $m_{t\_avr,i,j}$ and the corrected capacitance values $C_{m,i,j}$ calculated from (22) by the microcontroller were sent to the personal computer.

These results of scaling of the capacitive sensor interface are presented in Fig. 14a, and the relative errors of capacitance determination based on (22) - in Fig. 14b. The relative errors are in a range of about -0.075% to 0.081% and - as we can see - they are burdened with an “offset” assigned to a given value of C_x. These offset values are constants for particular $C_{x,i}$ values, which has been experimentally confirmed by 8192 measurements (128 series consisting of 64 measurements each) for a given $C_{x,i}$ value. Therefore, further software correction of this error is possible.

Dynamic correction is proposed. In this case, three sets of coefficients ($\{\beta_{corr,i}\}_{i = 1, .., 9}$, $\{C_{corr,i}\}_{i = 1, .., 9}$ and $\{m_{t\_max,i}\}_{i = 1, .., 9}$) are generated and placed in the program code of the

microcontroller, where the first two are calculated from (26) and (27), and the last set consists of the maximum values of measured times $m_{t\_max,i} = \max_j \{m_{t\_avr,i,j}\}$ for each $C_{x,i}$ value:

$$\beta_{corr,i} = \frac{C_{x,i+1} - C_{x,i}}{m_{t\_mean,i+1} - m_{t\_mean,i}} \tag{26}$$

$$C_{corr,i} = \frac{m_{t\_mean,i+1} \cdot C_{x,i} - m_{t\_mean,i} \cdot C_{x,i+1}}{m_{t\_mean,i+1} - m_{t\_mean,i}}, \tag{27}$$

where $m_{t\_mean,i} = \left(\min_j \{m_{t\_avr,i,j}\} + \max_j \{m_{t\_avr,i,j}\} \right) / 2$,

Next, final values of β_{corr} and C_{corr}, used to calculate $C_{m\_corr,i,j}$ based on (22) for the measured value of $m_{t\_avr,i,j}$, are selected from these sets according to the algorithm presented in Listing 1 in the form of a very simple code which is only a few lines long.

```
i = I - 1;
while(m_t < m_t_max(i)) {
        Beta_corr = Beta_corr_i(i);
        C_corr = C_corr_i(i);
        i--; }
```

Listing. 1. The code implementing the procedure for selecting values β_{corr} and C_{corr}

Thanks to this, as shown in Fig. 14c, the obtained relative errors are in a range of -0.057% to 0.049%, which for the assumed capacitance range of the capacitive sensor corresponds to a capacitance measurement accuracy ΔC_m of less than 0.1 pF. Hence, this value is twice as good as the required value of 1% for RH resolution [6].

5. Comparison of the methods with the state of the art

The new method proposed in the paper, based on a versatile direct sensor-to-microcontroller interface circuit (VIC), is compared with a 3-point calibration technique (3-PCT) [6], because both methods are based on direct sensor-to-microcontroller interface circuits dedicated to capacitive sensors. The first method is dedicated to capacitive sensors which can be grounded, whereas the second one - to floating capacitive sensors. The results of comparison are included in Table 1, in which the interface circuit complexity, the maximum measurement relative error, and the full measurement time of a single measurement are compared.

Table 1. Comparison of the proposed VIC method with the 3-point calibration technique.

Method	**Interface circuit**	**Built-in peripherals**	**Max. relative error**	**Measurement time**
VIC	R_r	2 I/O pins, timer, AC, DAC (internal voltage reference source)	0.081% (0.057%) for M = 64 (1)	$\boldsymbol{t_{Cm} + \xi t_{Cm}}$
3-PCT	R_d, R_i, C_c, (C_{off})	3 I/O pins, timer	0.036% for M = 100 (2)	$t_{cCx} + t_{cCc} + t_{cCoff} + t_{Cx} + t_{Cc} + t_{Coff}$

where:

t_{cCx}, t_{cCc}, t_{cCoff} – times of charging C_x, C_c,and C_{off} to the value of V_{cc} via the internal resistances of microcontroller pins,, respectively,

t_{Cx}, t_{Cc}, t_{Coff} – times of discharging C_x, C_c,and C_{off} via R_d and R_i to V_{TL} , respectively, V_{TL} – a low threshold voltage for CMOS digital circuits (0.4 V for ATXmega32A4 [23]),

t_{Cm} – time of charging C_x via R_r to V_m, where $V_{cc} > V_m > V_{TL}$.

Notes:

(1) for a range of C_x from 100 pF to 225 pF.

(2) for a range of C_x from 149 pF to 206 pF.

An important advantage of the proposed approach, as we can see in Table 1, is the simplicity of the interface circuit, as described in [23]. We use only one component, i.e. the reference resistor R_r, instead of three components: two reference resistors R_d, R_i and one capacitor C_c. This was made possible by transferring "the complexity of the interface circuit" inside the microcontroller. We use three peripherals (AC, DAC and timer) instead of one (timer), which is not a problem, because they are already built in the microcontroller.

The maximum relative error is at a similar level for both methods. It can be assumed that by increasing M from 64 to 100, a similar relative error value would be obtained (see Fig. 12).

It should be underlined that the most important advantage of the proposed method is several times shorter its measurement time. The capacitances of the interface circuit are charged to V_{cc} (the durations of charging are very short and can be neglected) and discharged to V_{TL} three times for the 3-PCT method. But, for the VIC method, capacitance C_x is charged only once to a level $V_m = \alpha \cdot V_{cc}$ ($\alpha \approx 0.5\ V_{cc}$) which additionally reduces the energy consumed by the interface circuit. The discharging times for the 3-PCT method and the charging time for the VIC method should be similar for maintaining the same measurement resolution of the timer of microcontroller. Hence, summarizing, we can say that the measurement time of the VIC method is about three times shorter

than that for the 3-PCT method, which also lowers energy consumption – e.g., among others, the core processor of the microcontroller works shorter (there is no information about using low-power consumption techniques in [6]).

6. Conclusions

In the paper there is presented a new time-domain measurement method for determining capacitance values of capacitive sensors, dedicated, among others, to capacitive relative humidity sensors and to grounded capacitive sensors, based on a versatile direct sensor-to-microcontroller interface circuit. The method is designed for microcontrollers equipped with internal analog comparators and precise voltage reference sources, e.g. digital-to-analog converters. If a microcontroller does not have such a reference source, we can use a resistive divider attached to the negative input of the analog comparator instead. The interface circuit is very simple – it consists only of a reference resistor and a capacitive sensor, both acting as a voltage divider. It was adjusted to measure capacitances of tens and hundreds of pF.

An example of complete application in the form of a prototype based on an 8-bit ATXmega32A4 microcontroller has been developed and experimentally tested. The experiments confirmed that relative errors of capacitance value determination for a capacitive sensor are less than ±0.06%, which corresponds to a capacitance measurement accuracy less than 0.1 pF for a range of the measured capacity values from 100 pF to 225 pF, which in turn corresponds to at least a 0.5% relative humidity resolution for commercial capacitive RH sensors (e.g. TE Connectivity HS1101LF and Philips H1).

It should be underlined that the measurement procedure based on the proposed method is short, so that we obtain a low-cost and low-power solution for a smart capacitance sensor. Also, the determination of capacitive sensor capacitance values is very simple – it is based on a linear polynomial. Hence, the software implementing this method needs low computing power and little space in program and data memories. Thus it can be implemented even in 8-bit microcontrollers, as it was described in the paper. Another advantage of the proposed solution is its capability to extend the functionality of existing microcontroller systems with direct sensor-to-microcontroller interface circuits in a cheap and simple way.

The proposed solution can also be used to design battery-powered compact smart capacitive sensors employed in wireless sensor networks, e.g. networks applying the protocols based on the IEEE 802.15.4 standard.

References

[1] T. A. Blank, L. P. Eksperiandova, K. N. Belikov, Recent trends of ceramic humidity sensors development: A review, Sensors and Actuators B 228 (2016) 416–442.

[2] Z. M. Rittersm, Recent achievements in miniaturised humidity sensors - a review of transduction techniques, Sensors and Actuators A 96 7 (2002) 196-210.

[3] M. Dokmeci, K. Najafi, A High-Sensitivity Polyimide Capacitive Relative Humidity Sensor for Monitoring Anodically Bonded Hermetic Micropackages, Journal of Microelectromechanical Systems, 10 (2) (2001) 197-204.

[4] Y. Kim, B. Jung, H. Lee, H. Kim, K. Lee, H. Park, Capacitive humidity sensor design based on anodic aluminum oxide, Sensors and Actuators B 141 (2009) 441–446.

[5] A. Rivadeneyra, J. Fernandez-Salmeron, M. Agudo-Acemel, J. A. Lopez-Villanueva, L. F. Capitan-Vallvey, A. J. Palmac, Printed electrodes structures as capacitive humidity sensors: A comparison, Sensors and Actuators A 244 (2016) 56–65.

[6] F. Reverter, O. Casas, Direct interface circuit for capacitive humidity sensors, Sensors and Actuators A 143 (2008) 315–322.

[7] TE Connectivity Ltd., HS1101LF Relative Humidity Sensor, SENSOR SOLUTIONS /// HS1101LF HPC052_J (2015).

[8] Philips Components, Humidity sensor 2322 691 90001 Product specification, (1996).

[9] G. Tuna, V. C . Gungor, Ch2 – Energy harvesting and battery technologies for powering wireless sensor networks, Industrial Wireless Sensor Networks. Woodhead Publishing (2016), 25-38.

[10] M. Kuorilehto, M. Kohvakka, J. Suhonen, P. Hamalainen, M. Hannikainen T. D. Hamalainen, Ultra-low energy wireless sensor networks in practice, John Wiley & Sons, Ltd., Great Britain, 2007.

[11] F. Reverter, M. Gasulla, R. Pallàs-Areny, Analysis of power-supply interference effects on direct sensor-to-microcontroller interfaces, IEEE Transactions on Instrumentation and Measurement 56 (1) (2007) 171-177.

[12] F. Reverter, The Art of Directly Interfacing Sensors to Microcontrollers, Journal of Low Power Electronics and Applications (2) (2012) 265-281.

[13] F. Reverter, Ò. Casas, Interfacing differential resistive sensors to microcontrollers: A direct approach, IEEE Transactions on Instrumentation and Measurement 58 (10) (2009) 3405-3410.

[14] F. Reverter, O. Casas, A microcontroller-based interface circuit for lossy capacitive sensors, Measurement Science Technology 21 (2010) 065203, 1-8.

[15] F. Reverter, O. Casas, Interfacing Differential Capacitive Sensors to Microcontrollers: A Direct Approach, IEEE Transactions on Instrumentation and Measurement 59 (2010) 2763-2769.

[16] J. E. Gaitán-Pitre, M. Gasulla, R. Pallàs-Areny, Analysis of a Direct Interface Circuit for Capacitive Sensors, IEEE Transactions on Instrumentation and Measurement 58 (9) (2009) 2932-2937.

[17] J. Pelegrí-Sebastiá, E. García-Breijo, J. Ibáñez, T. Sogorb, N. Laguarda-Miro, J. Garrigues, Low-Cost Capacitive Humidity Sensor for Application Within Flexible RFID Labels Based on Microcontroller Systems, IEEE Transactions on Instrumentation and Measurement 61 (2) (2012) 545-553.

[18] O. Lopez-Lapeña, E. Serrano-Finetti and O. Casas, Calibration-less direct capacitor-to-microcontroller interface, Electronics Letters 52 (4) (2016) 289–291.

[19] Z. Kokolanski, J. Jordana, M. Gasulla, V. Dimcev, F. Reverter, Direct inductive sensor-to-microcontroller interface circuit, Sensors and Actuators A 224 (2015) 185–191.

[20] Z. Kokolanski, F. Reverter, C. Gavrovski, V. Dimcev, Improving the resolution in direct inductive sensor-to-microcontroller interface, Annual Journal Of Electronics (2015) 135-138.

[21] Z. Czaja, A microcontroller system for measurement of three independent components in impedance sensors using a single square pulse, Sensors and Actuators A 173 (2012) 284-292.

[22] Z. Czaja, An implementation of a compact smart resistive sensor based on a microcontroller with an internal ADC, Metrology and Measurement Systems 23 (2016) 255-238.

[23] Z. Czaja, Time-domain measurement methods for R, L and C sensors based on a versatile direct sensor-to-microcontroller interface circuit, Sensors and Actuators A 274 (2018) 199–210.

[24] F. Reverter, X. Li, G. C M Meijer, Stability and accuracy of active shielding for grounded capacitive sensors, Measurement Science and Technology 17 (2006) 2884–2890.

[25] F. Reverter, X. Li, G. C M Meijer, A novel interface circuit for grounded capacitive sensors with feedforward-based active shielding, Measurement Science and Technology 19 (2008) 025202 (5pp).

[26] Atmel Corporation, 8-bit XMEGA A Microcontroller, XMEGA AU MANUAL, (2013), Available at: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-8331-8-and-16-bit-AVR-Microcontroller-XMEGA-AU_Manual.pdf.

[27] Atmel Corporation, 8/16-bit AVR XMEGA A4 Microcontroller. ATxmega128A4, ATxmega64A4, ATxmega32A4, ATxmega16A4, (2013), Available at: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-8069-8-and-16-bit-AVR-AMEGA-A4-Microcontrollers_Datasheet.pdf.

[28] Microchip Technology Inc., AVR1300: Using the Atmel AVR XMEGA ADC, (2017), Available at: http://ww1.microchip.com/downloads/en/Appnotes/00002535A.pdf.

[29] F. Reverter, R. Pallàs-Areny, Effective number of resolution bits in direct sensor-to-microcontroller interfaces, Measurement Science and Technology 15 (2004) 2157–2162.

[30] F. Reverter, R. Pallàs-Areny, Uncertainty reduction techniques in microcontroller-based time measurements, Sensors and Actuators A 127 (2006) 74–79.

[31] K. Kolikov, G. Krastevy, Y. Epitropov, A. Corlat, Analytically determining of the relative inaccuracy (error) of indirectly measurable variable and dimensionless scale characterizing quality of the experiment, Computer Science Journal of Moldova 20 (58) (2012) 15-32.

[32] I. Farrance, R. Frenkel, Uncertainty of measurement: A review of the rules for calculating uncertainty components through functional relationships, The Clinical Biochemist Reviews 33 (2012), 49-75.

[33] Agilent Technologies, Agilent 34410A/11A 6 1/2 digit multimeter user's guide (2012).

Figures captions

Fig. 1. Scheme of a versatile direct sensor-to-microcontroller interface circuit.

Fig. 2. Block scheme of the compact smart capacitive sensor based on the ATxmega32A4 microcontroller.

Fig. 3. Timings of the stimulation signal $v_{in}(t)$ of the sensor interface circuit and its voltage time response $v_{out}(t)$ for V_{in} = 3.3 V, R_r = 10 MΩ, C_x = 100 nF.

Fig. 4. The flowchart of the algorithm of the measurement procedure.

Fig. 5. Circuit model of a versatile direct capacitive sensor-to-microcontroller interface circuit.

Fig. 6. Simplified circuit model of the interface circuit from Fig. 5.

Fig. 7. Graph of the inaccuracy of the DAC conversion $\delta\alpha$ as a function of the voltage ratio $\alpha = V_m/V_{cc.}$

Fig. 8. Graph of the trigger inaccuracy of the AC $\delta m_{trigger}$ as a function of noise (disturbances) v_{noise} at the input pin of the AC.

Fig. 9. (a) Graph of t_m as a function of C_x values for the capacitive sensor. (b) Scaling of the capacitive sensor. (c) Scaling of the capacitive sensor after software correction.

Fig. 10. (a) Relative errors of the capacitance value determination after software correction. (b) Measurement capacitance accuracy ΔC_m as a function of C_x values.

Fig. 11. Flowchart of the algorithm of the *M*-procedure.

Fig. 12. Discharging path of a versatile direct capacitive sensor-to-microcontroller interface circuit: a) circuit model, b) simplified circuit model.

Fig. 13. Graph of a measurement capacitance accuracy ΔC_m as a function of the number M of the time t_m measurement procedure repetitions for C_x = 100.09 pF.

Fig. 14. (a) Scaling of the capacitive sensor for the M-procedure. (b) Relative errors of the capacitance value determination after software correction for the M-procedure. (c) Relative errors of the capacitance value determination after dynamic software correction for the M-procedure.

Table captions

Table. 1. Comparison of the proposed VIC method with the 3-point calibration technique method.

Figure 1

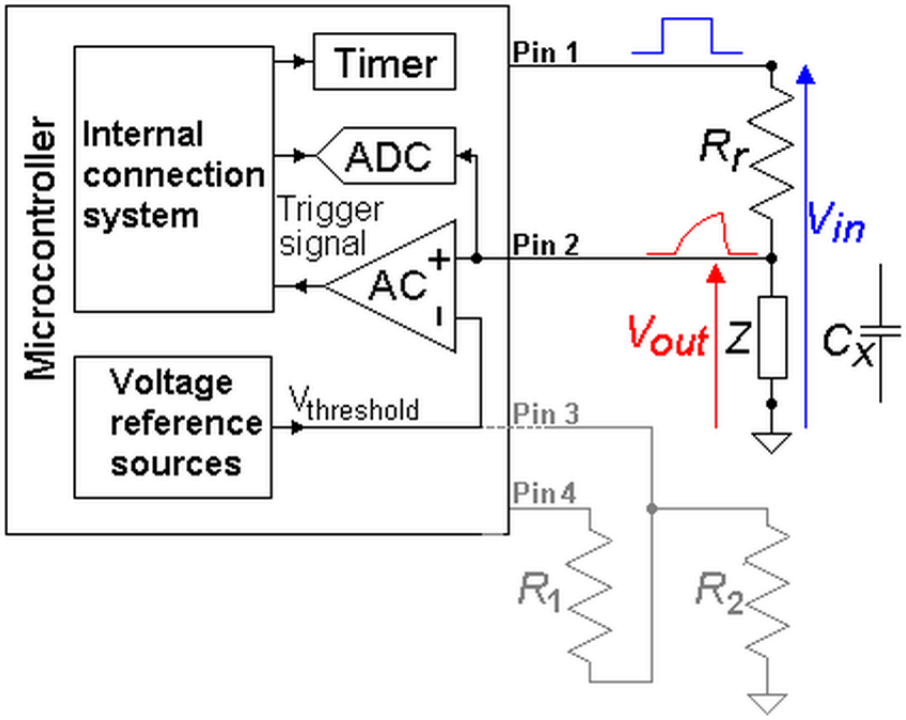

Figure 2

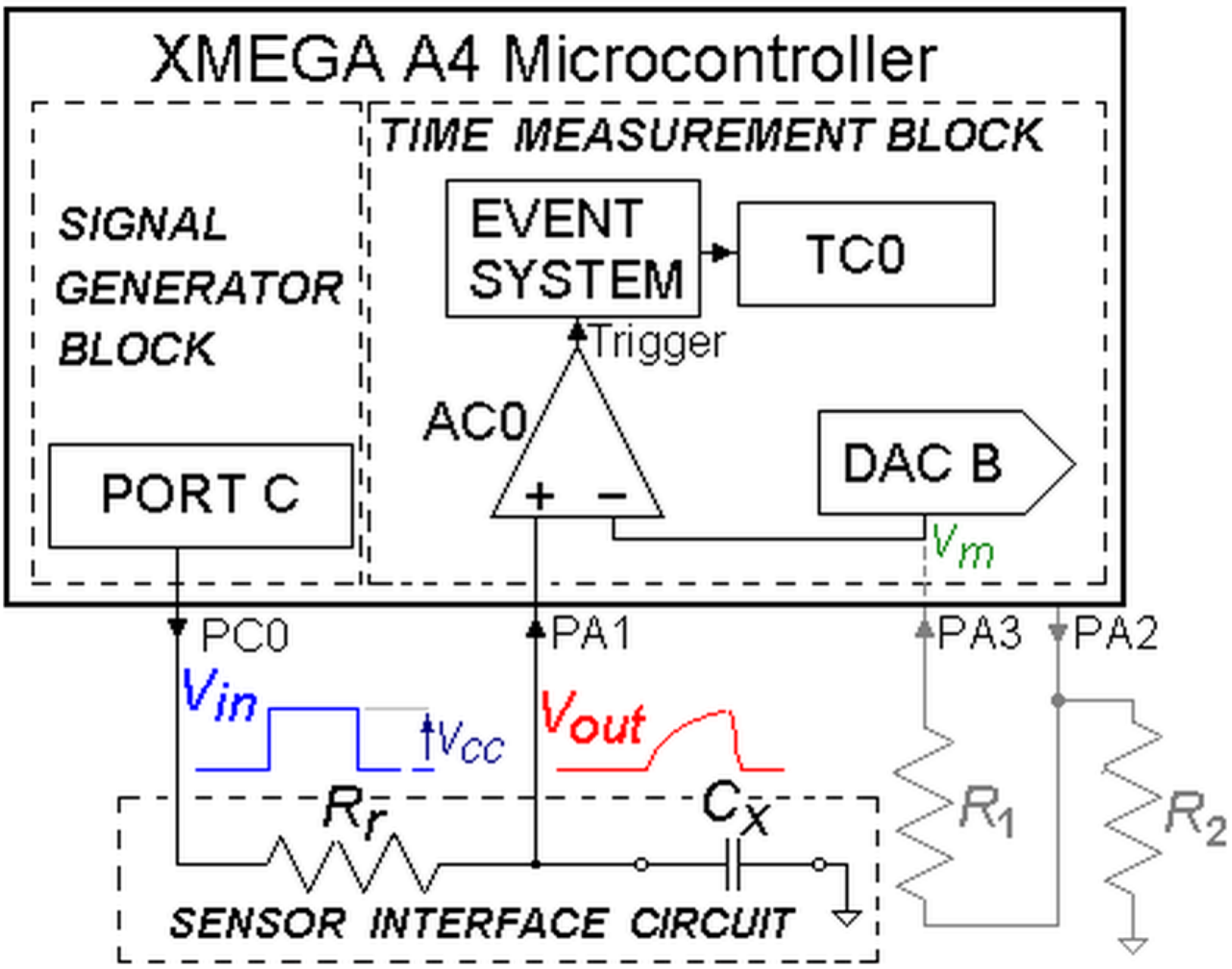

Figure 3

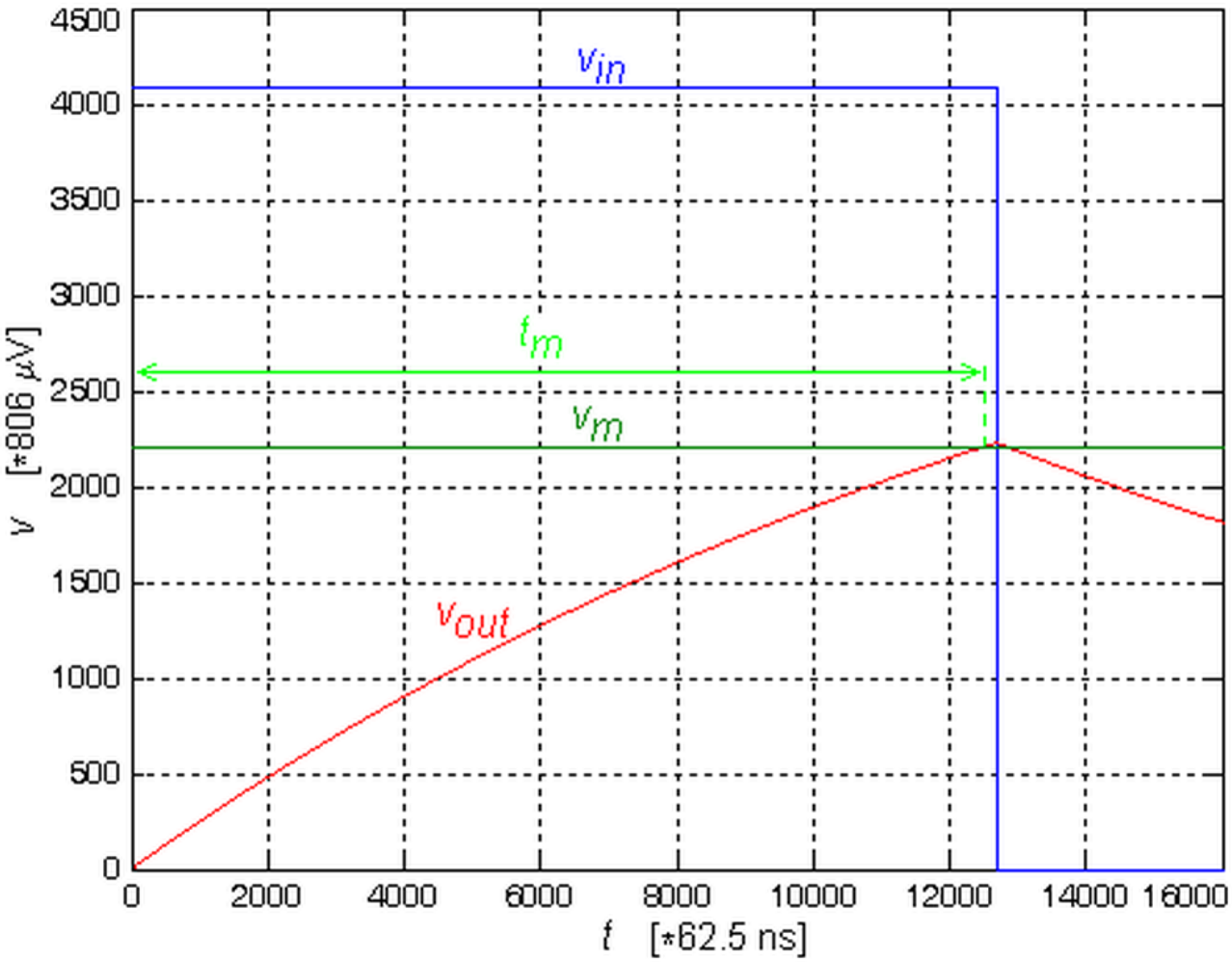

Figure 4

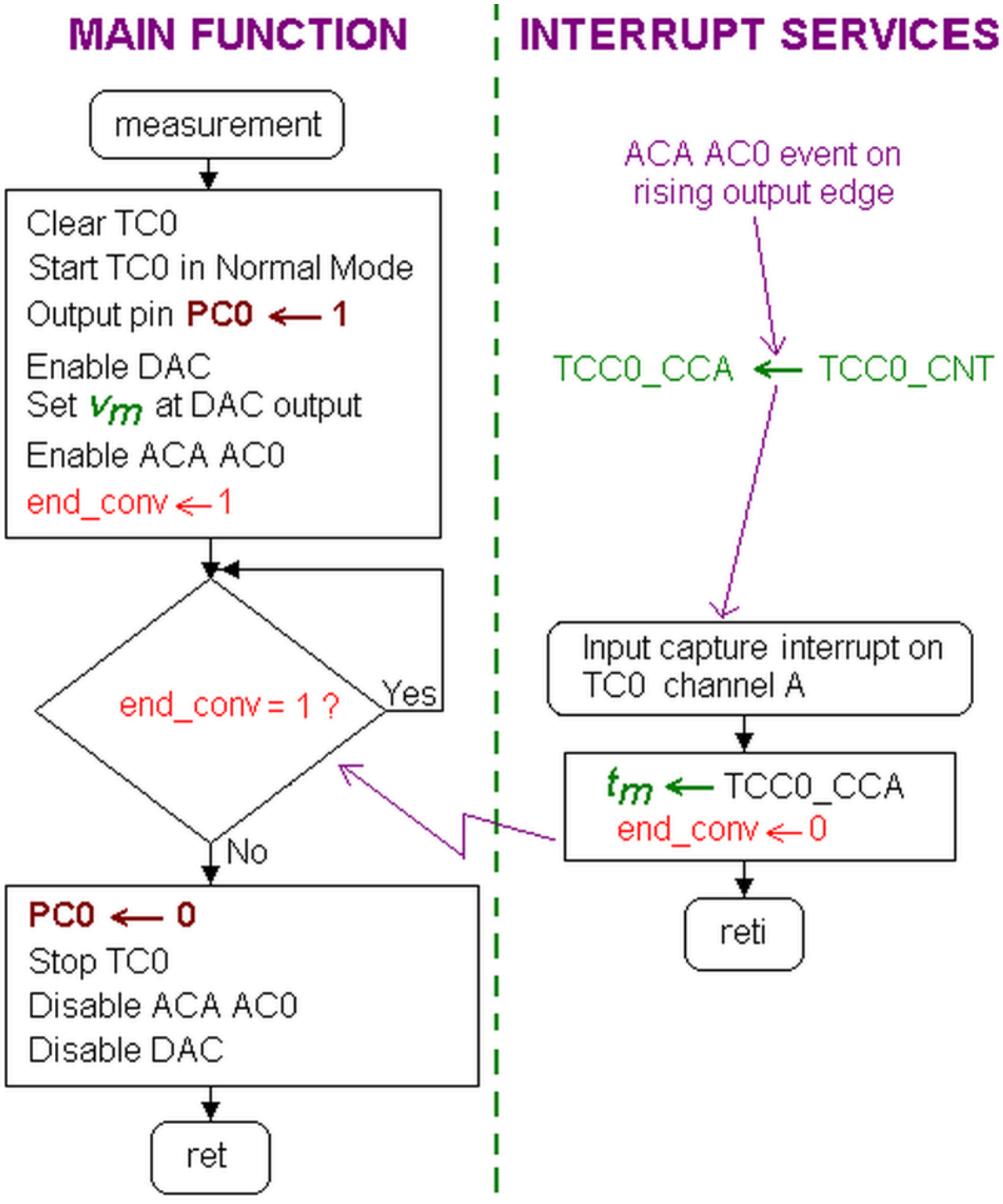

Figure 5

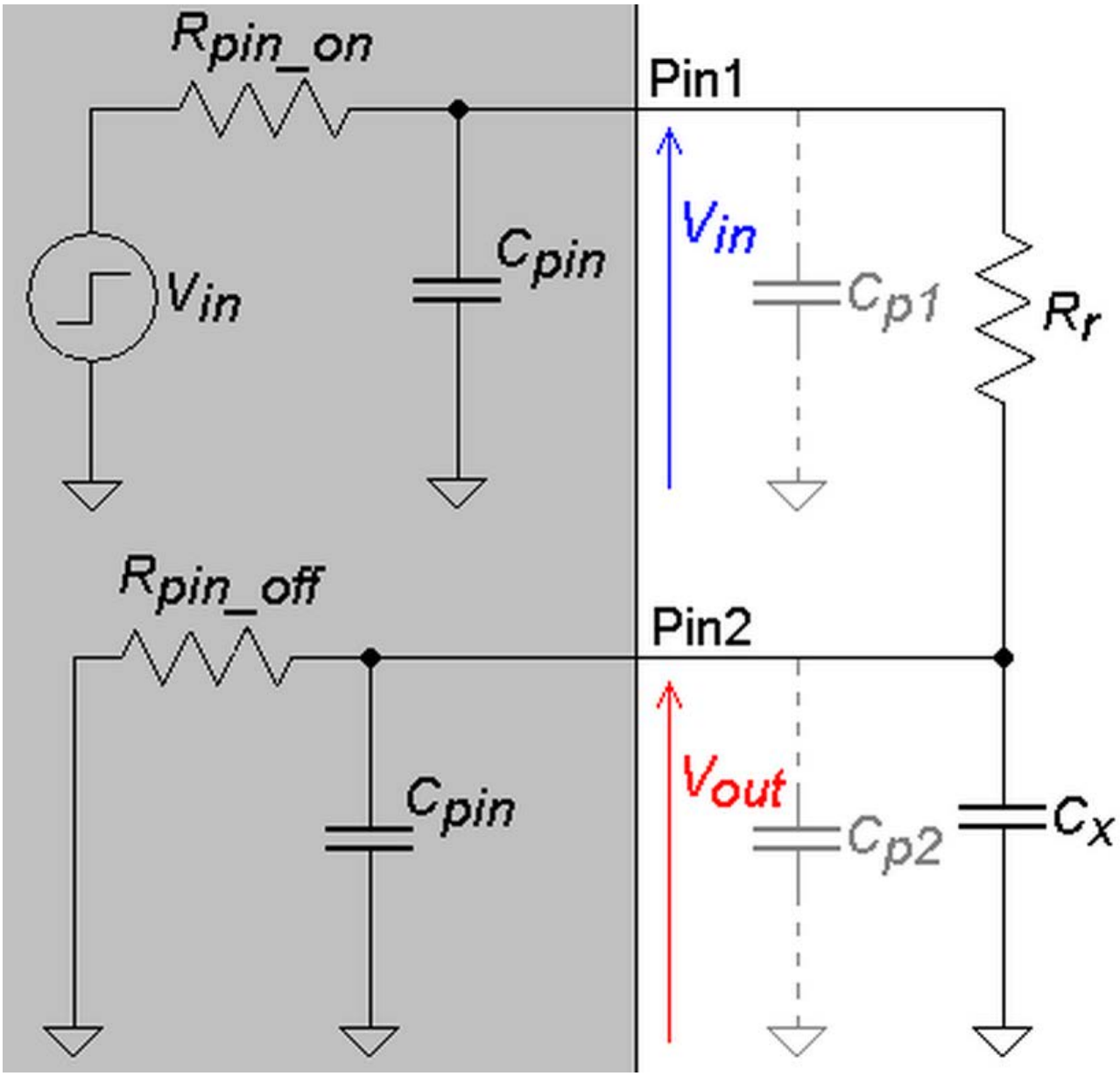

Figure 6

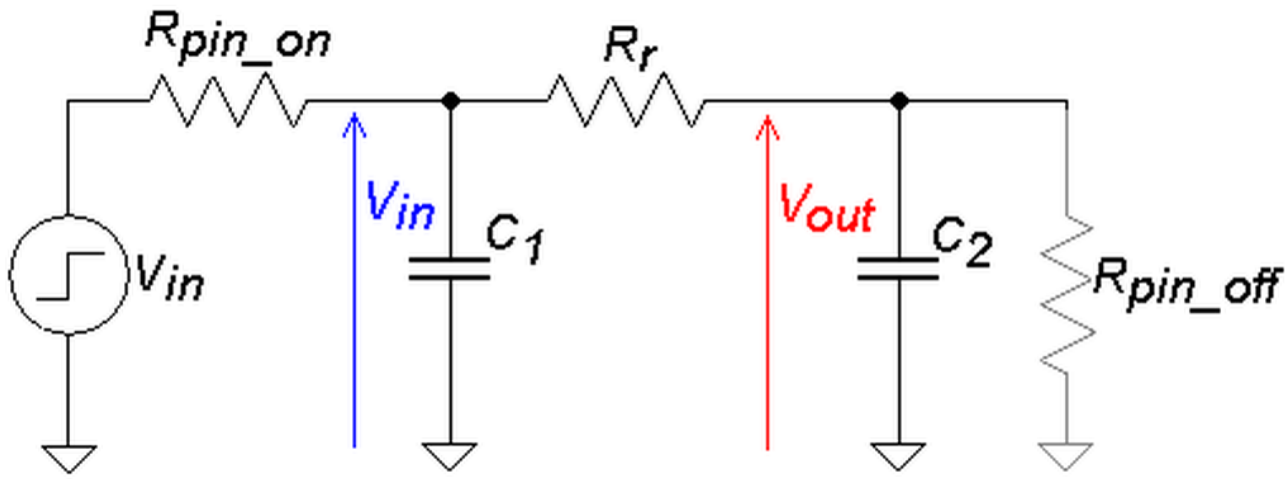

Figure 7

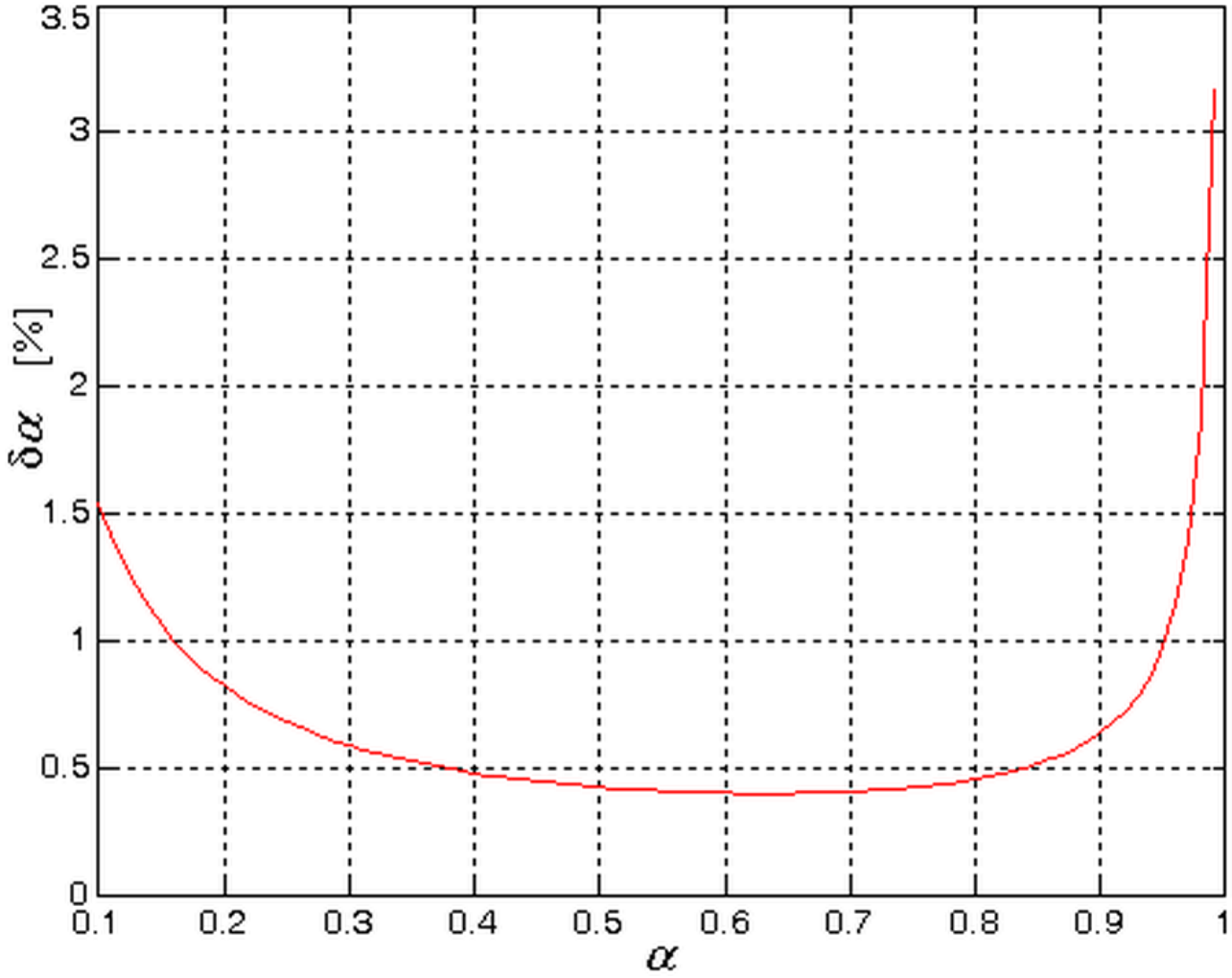

Figure 8

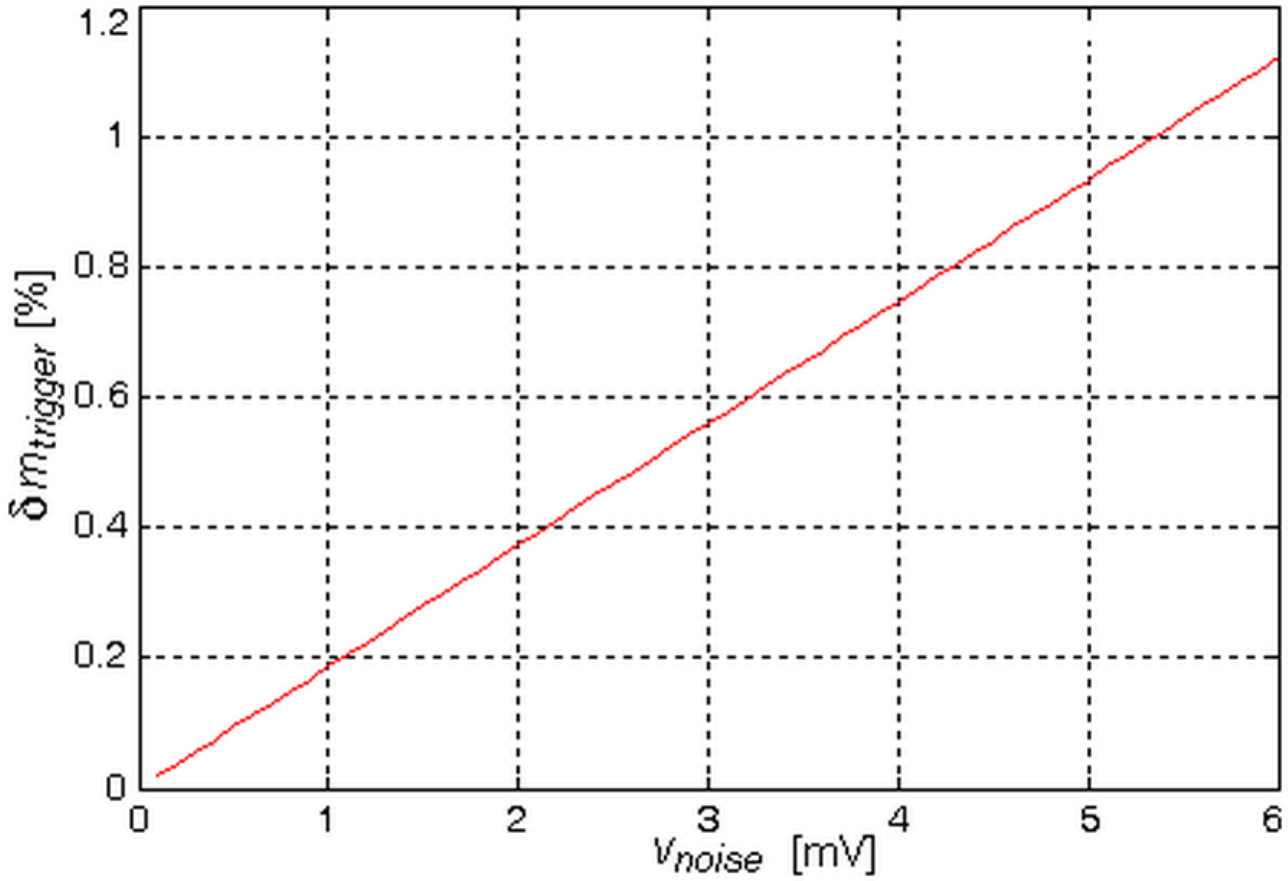

Figure 9

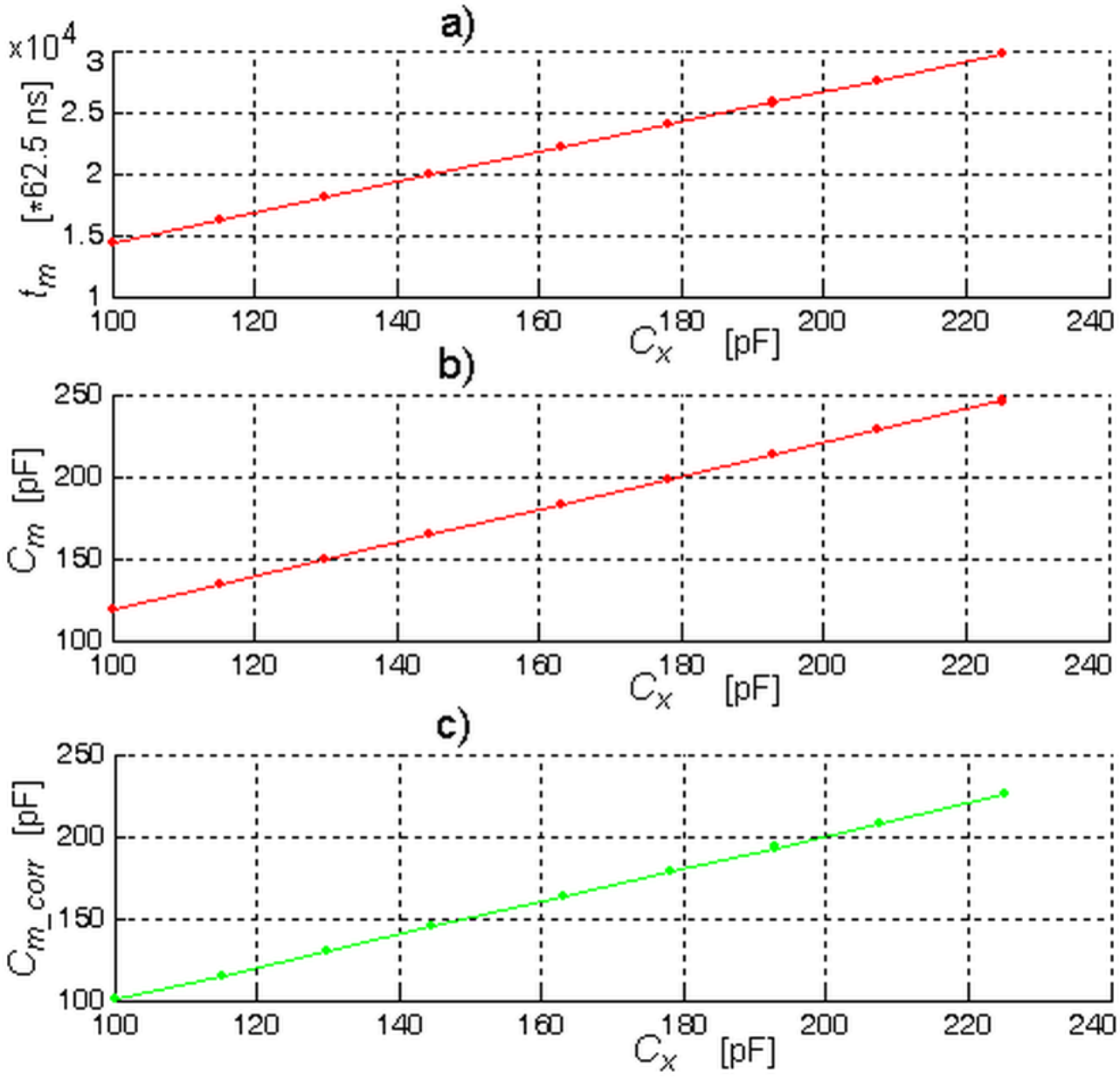

Figure 10

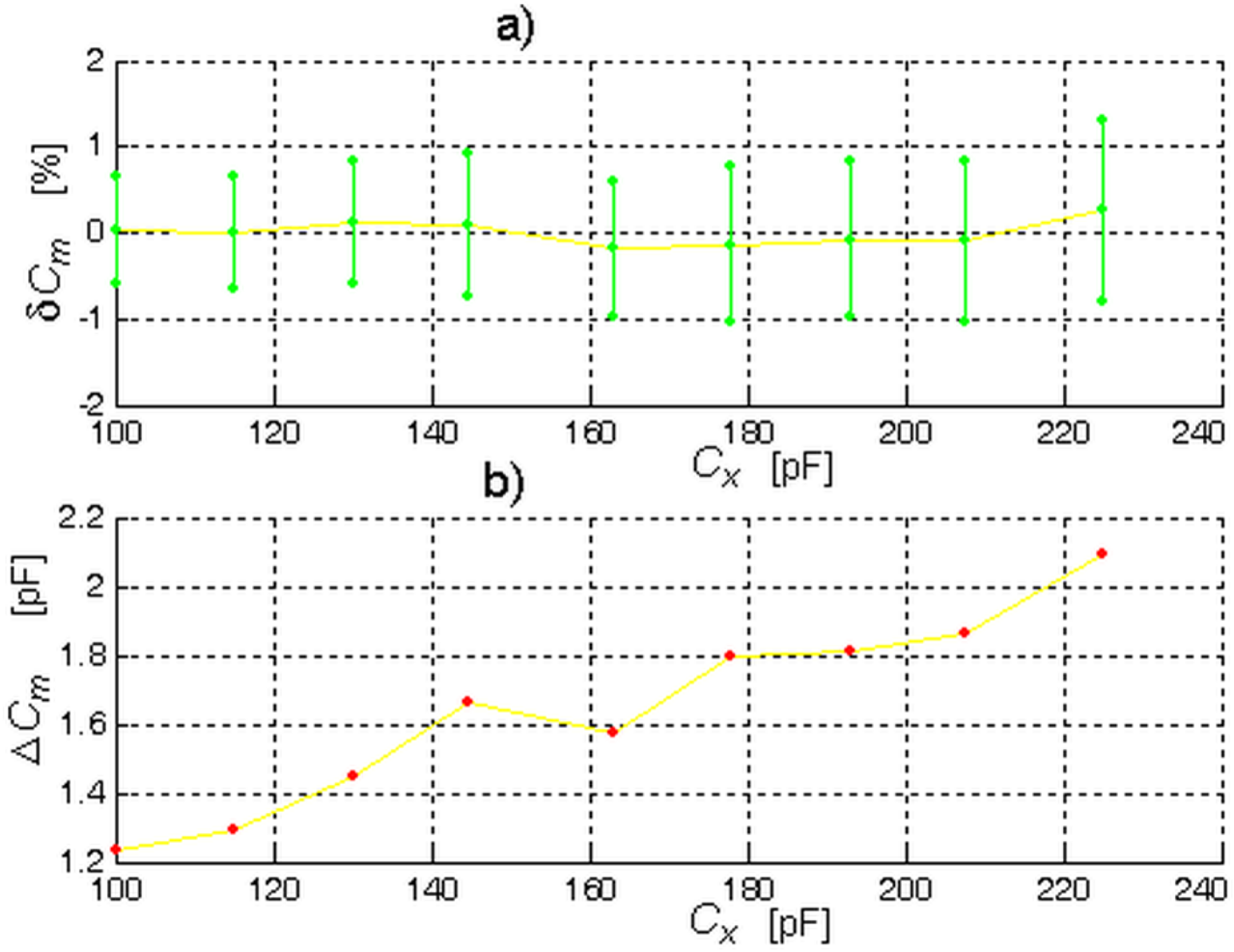

Figure 11

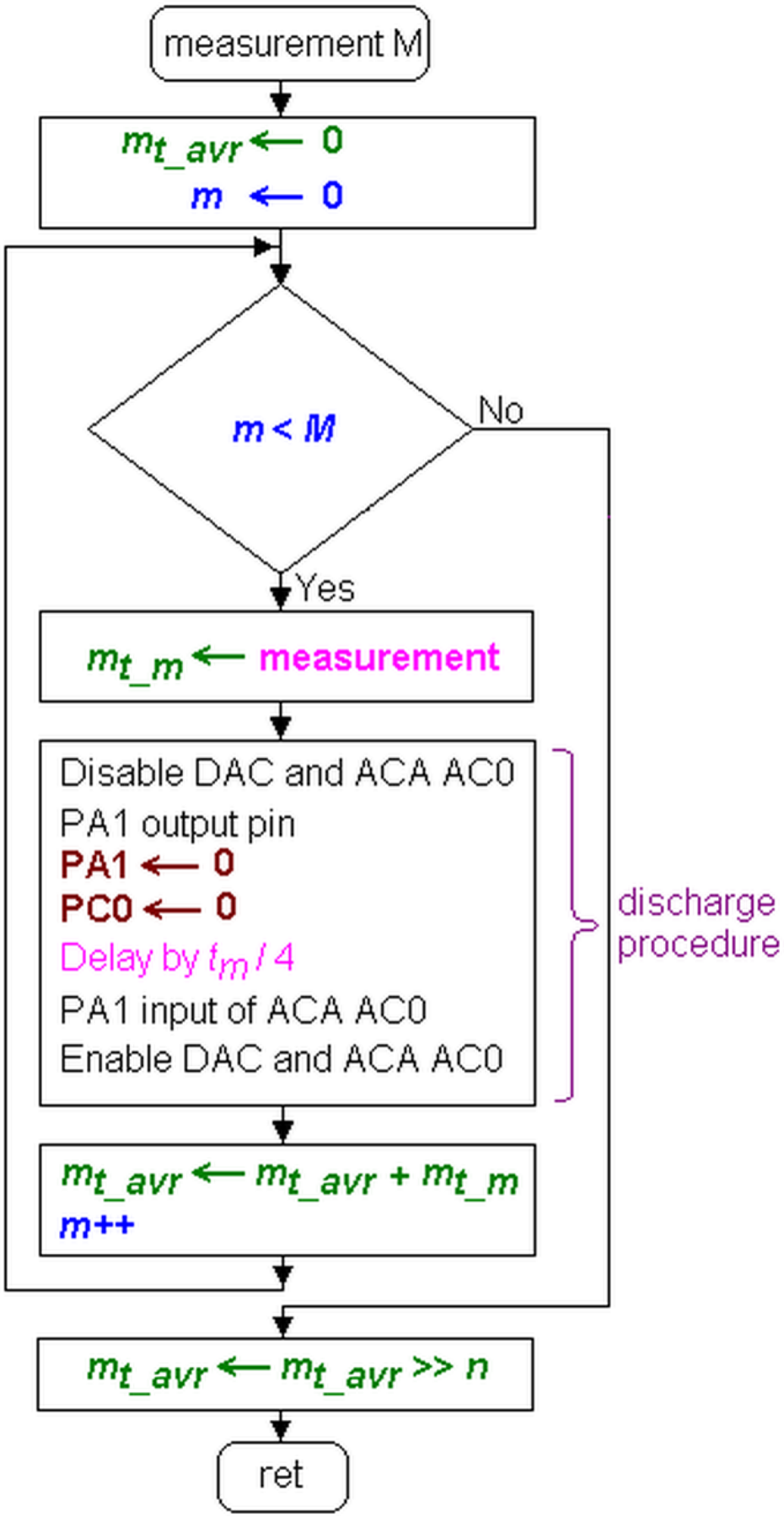

Figure 12a

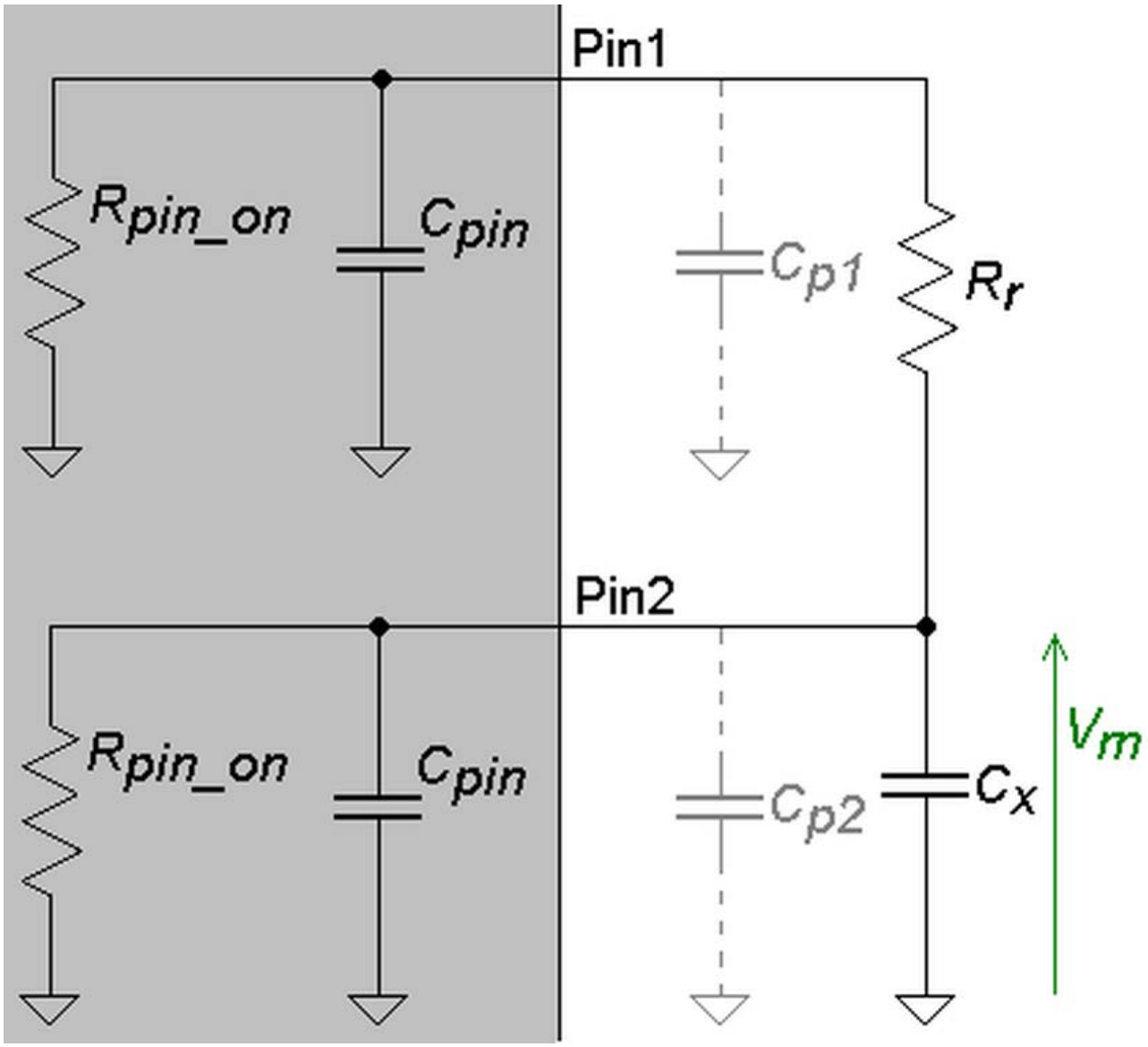

Figure 12a

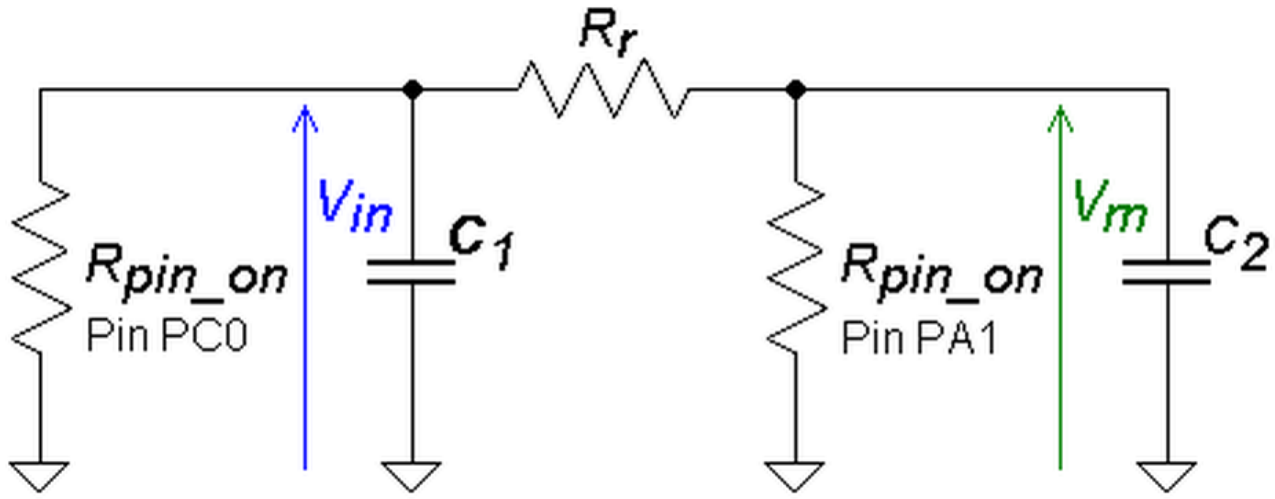

Figure 13

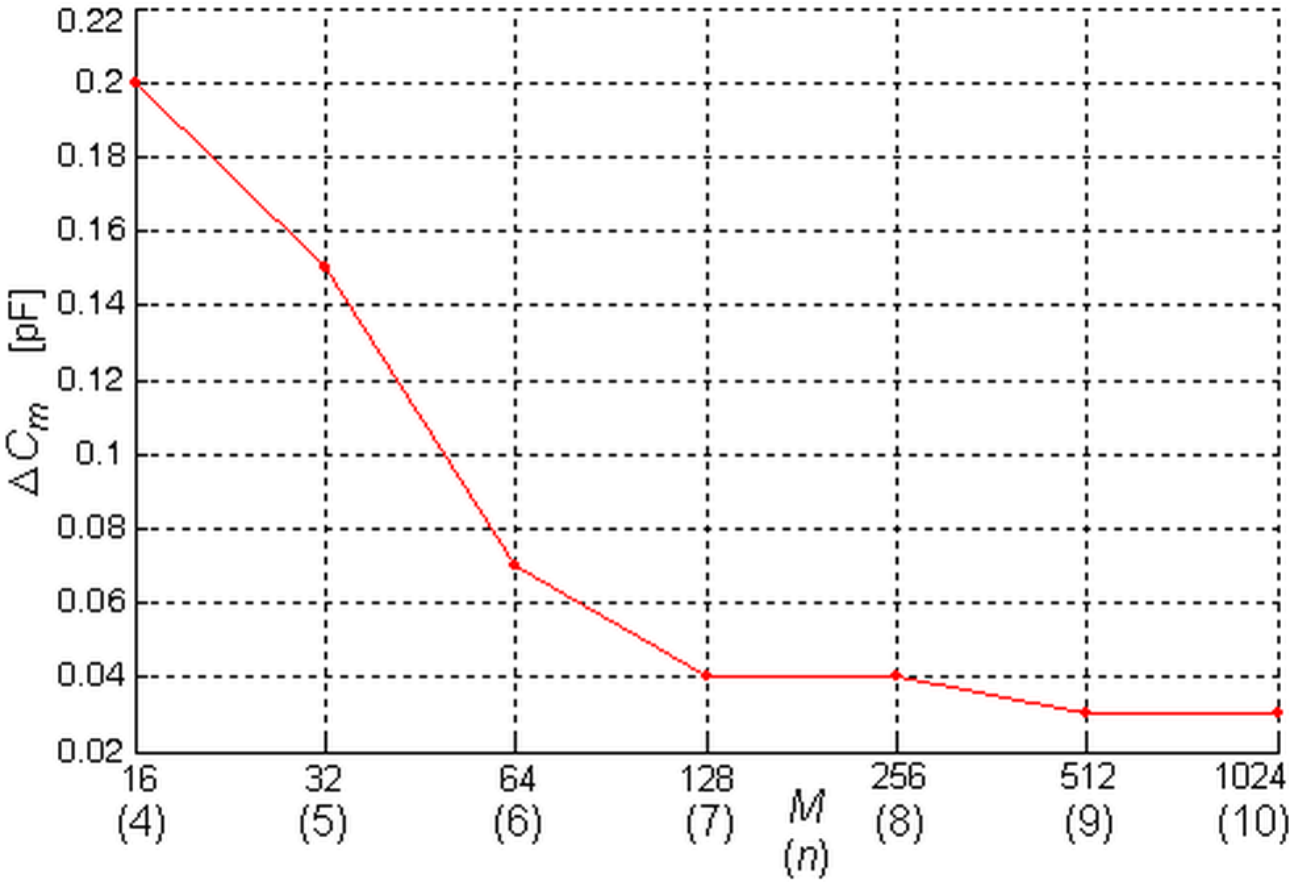

Figure 14