Advanced Techniques for Directly Interfacing Resistive Sensors to Digital Systems

Ferran Reverter, Fernando Vidal-Verdú and José A. Hidalgo-Lopez

Abstract This chapter reviews advanced techniques for the direct connection of resistive sensors to digital systems without using any analogue circuit, such as an amplifier or an analogue-to-digital converter, in the signal path. The sensor electronic interfaces proposed herein rely on the following operating principle: the digital system measures through an embedded digital timer the charging/discharging time of an RC circuit formed by the resistive sensor and a known capacitor. The chapter first explains how resistive sensors with a single, differential or bridge topology can be directly measured using a low-cost microcontroller. The uncertainty sources involved in the measurement (such as the mismatch of the internal resistances, quantisation and trigger noise) and the performance in some applications are reported. Next, the chapter deals with the direct connection of resistive sensor arrays to field-programmable gate arrays, where different resistances of the array are measured in parallel through a set of timers running simultaneously. The new uncertainty sources (mainly, crosstalk) and the applications are also reported. Although the proposed sensor interfaces are quite simple in terms of operating principle, their linearity and resolution are quite remarkable provided that the design rules indicated along this chapter are followed.

F. Reverter (✉)
Universitat Politècnica de Catalunya - BarcelonaTech, Barcelona, Spain
e-mail: ferran.reverter@upc.edu

F. Vidal-Verdú · J.A. Hidalgo-Lopez
Universidad de Málaga - AndalucíaTech, Málaga, Spain
e-mail: fvidal@uma.es

J.A. Hidalgo-Lopez
e-mail: jahidalgo@uma.es

B. George et al. (eds.), *Advanced Interfacing Techniques for Sensors*,
Smart Sensors, Measurement and Instrumentation 25,
DOI 10.1007/978-3-319-55369-6_4

1 Introduction

In the society of the 21st century, many people have smart home appliances inside smart buildings located in smart cities whose streets are full of smart cars. Almost everything is getting smart thanks to the proliferation of *information and communication technology* and the deployment of technologies such as *wireless sensor networks* and the *internet of things*. To become smart, it is essential in the first place to monitor through sensors what is happening in and/or around the smart *thing*. The data collected is processed and then a smart decision is taken with the aim of improving the safety, efficiency, sustainability, mobility, etc. of the smart *thing* and, hence, the people's quality of life.

Real-time monitoring systems employ sensors to acquire information, the same as human beings use their senses. The information acquired can be very diverse, for instance: the carbon monoxide (CO) concentration in air in a smart city, the tire pressure in a smart car, the vibration level in a smart building, or the laundry weight in a smart washing machine. The magnitude of the measurand (e.g. CO concentration, pressure, vibration and weight) determines the magnitude of the electrical signal (e.g. resistance, capacitance, voltage or current) provided at the sensor output. Anyhow, such an electrical signal is generally of low amplitude and carries some noise and, therefore, an electronic interface is required between the sensor and the processing digital system so as to correctly extract the information of interest.

A classical block diagram of a sensor electronic interface is shown in Fig. 1 [1]. The sensor output signal is first processed in the analogue domain by a signal conditioning circuit that generally relies on operational amplifiers (OpAmp). The main functions of this block are level shifting and amplification so as to match the sensor output span to the input span of the ensuing analogue-to-digital converter (ADC) and, hence, to make good use of the ADC dynamic range. Other common tasks of the signal conditioning circuit are: sensor output-to-voltage conversion, filtering, linearization and/or demodulation. The resulting analogue signal is then digitized via the ADC. Finally, a digital system acquires, stores, processes, controls, communicates (to other devices or systems) and/or displays the digital value with information about the measurand. Nowadays, the most popular digital systems are microcontrollers (μC) and Field-Programmable Gate Arrays (FPGA).

The sensor electronic interface shown in Fig. 1 can be implemented in various ways, for example: (i) each block has its own integrated circuit (IC) and then those are interconnected in a printed circuit board (PCB); or (ii) an application-specific IC

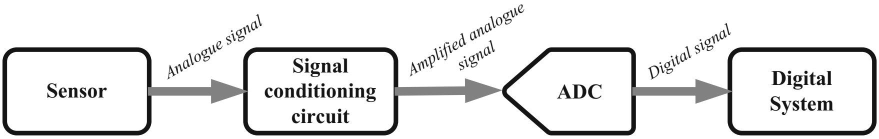


Fig. 1 Classical block diagram of a sensor electronic interface

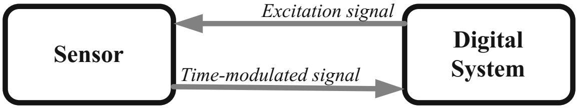


Fig. 2 Direct interface circuit

(ASIC) including the electronics of the four blocks shown in Fig. 1 is designed. Intermediate solutions are also offered by the main semiconductor companies through commercial ICs that include: (i) some signal conditioning circuit, the ADC and the digital system (e.g. MSC1210 from Texas Instruments, TI); (ii) the sensor, its signal conditioning circuit and the ADC (e.g. ADXL312 from Analog Devices, AD); and (iii) the signal conditioning circuit and the ADC to measure a specific type of sensor (e.g. ADS1232 from TI for bridge-type resistive sensors and AD7745 from AD for capacitive sensors). These chips including the sensor together with analogue and digital electronics are commonly known as integrated *smart sensors* [2].

An alternative approach to reading some sensors (e.g. resistive [3–5], capacitive [6–8], inductive [9, 10] and voltage-output [11] sensors) is shown in Fig. 2. This circuit topology is known as *direct interface circuit* since the sensor is directly connected to the digital system without using either the signal conditioning circuit or the ADC [12, 13]. The digital system excites the sensor to get a time-modulated signal that is directly measured in the digital domain through a digital timer embedded into the digital system. In comparison with the sensor electronic interface shown in Fig. 1, a direct interface circuit is simpler and needs fewer components. Actually, it can be implemented with a common general-purpose 8-bit µC which is a low-cost (say, 1 $) and low-power (say, about 1 mA in active mode and less than 1 µA in power-down mode [14]) device. Therefore, a direct interface circuit offers advantages in terms of cost, physical space and power consumption, which is of major interest, for instance, in autonomous sensors powered by either batteries or energy harvesters. Furthermore, as will be shown along this chapter, the performance of such circuits in terms of accuracy and resolution is quite remarkable taking into account their simplicity.

This chapter reviews most of the research work carried out about direct interface circuits for resistive sensor and is organized as follows. Section 2 describes the operating principle of such circuits. Section 3 explains how a µC can be applied to measure different topologies of resistive sensors. Section 4 does the same but using FPGA applied to resistive sensor arrays. Finally, Sect. 5 takes some conclusions and forecasts the future research work about this topic.

2 Operating Principle

Direct interface circuits for resistive sensors rely on measuring the charging or discharging time of an RC circuit, i.e. a circuit with a resistance (R) and a capacitance (C), as shown in Fig. 3a. The digital system excites the RC circuit and then

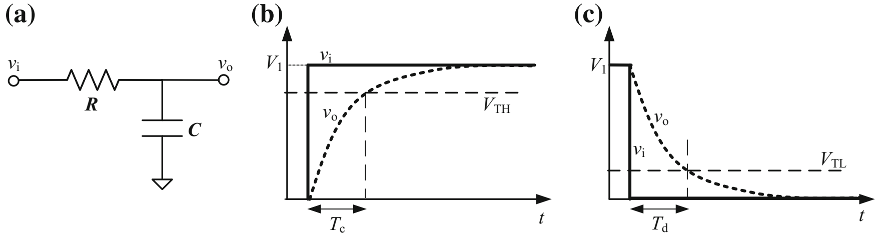


Fig. 3 **a** RC circuit; **b** measurement of the charging time; and **c** measurement of the discharging time

measures the time interval needed to charge or discharge the capacitance C to a given threshold voltage through the sensor resistance.

The basics of the operating principle for the measurement of the charging time and the discharging time are explained by means of Fig. 3b, c, respectively. In Fig. 3b, assuming C initially discharged, if a step of amplitude V_1 is applied to the input of the RC circuit, then the transient response of the output voltage is

$$v_o(t) = V_1\left(1 - e^{-\frac{t}{RC}}\right), \tag{1}$$

and the time required to charge C from 0 to a given high threshold voltage (V_{TH}) is

$$T_c = RC \ln\left(\frac{V_1}{V_1 - V_{TH}}\right), \tag{2}$$

which is proportional to R. On the other hand, in Fig. 3c, assuming C already charged to V_1, if a step towards ground is applied to the input, then the transient response of the output voltage is

$$v_o(t) = V_1 e^{-\frac{t}{RC}}, \tag{3}$$

and the time needed to discharge C from V_1 to a given low threshold voltage (V_{TL}) is

$$T_d = RC \ln\left(\frac{V_1}{V_{TL}}\right) \tag{4}$$

which again is proportional to R. Therefore, in an RC circuit, changes of resistance are proportionally converted to changes of time interval.

The RC circuit in Fig. 3a can be directly connected to a digital system using the circuit topology shown in Fig. 4a, where R has been replaced by R_x (i.e. a resistive sensor). Two input/output digital ports (pins 1 and P) are employed to excite the RC circuit and to monitor through a Schmitt trigger (ST) buffer embedded into Pin 1 the exponential charging or discharging voltage represented before in Fig. 3b, c,

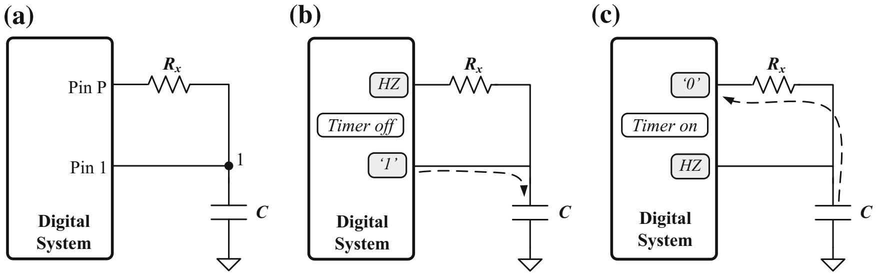


Fig. 4 **a** Basic topology of a direct interface circuit for a resistive sensor; **b** pin configuration during the charging stage; and **c** pin configuration during the discharging and measurement stage

respectively. This circuit can measure either the charging time or the discharging time, but the measurement of the latter is preferable since it has lower variability. This is because the discharging-time measurement uses the V_{TL} of the ST buffer, which is less noisy than the V_{TH} used for the charging-time measurement [15]. For this reason, the rest of the chapter always assumes that the direct interface circuit measures the sensor resistance through the discharging time.

The circuit in Fig. 4a involves two operation stages: charging stage, and discharging and measurement stage. During the charging stage, Pin 1 is set as an output providing a digital '1', whereas Pin P is set as an input offering high impedance (HZ), as shown in Fig. 4b. Therefore, the capacitor C is quickly charged to the analogue output voltage (V_1) corresponding to a digital '1', which is generally equal to the supply voltage (V_{DD}) of the digital system. During the discharging and measurement stage, Pin 1 is set as a HZ input and Pin P is set as an output providing a digital '0', as shown in Fig. 4c. Consequently, C is discharged towards ground through R_x while a digital timer (embedded into the digital system) measures the time interval required to do so. When the exponential discharging voltage crosses the V_{TL} of the ST buffer embedded into Pin 1, the timer is read and a digital number proportional to R_x (see Eq. (4)) is achieved.

3 Interfacing Resistive Sensors to Microcontrollers

The operating principle explained in Sect. 2 can be implemented by a μC to measure resistive sensors with a single, differential or bridge topology. Next, we discuss the main features of both the sensor and the μC, and then we explain how to join them to build a direct interface circuit. The uncertainty sources involved in the measurement and the application of the proposed circuits are also reported.

3.1 Sensor

In monitoring systems based on resistive sensors, the measurand directly or indirectly alters the electrical resistance (R) of a resistive element that can be modelled as

$$R = \rho \frac{l}{A}, \tag{5}$$

where ρ is the resistivity of the material, and l and A are the length and cross-sectional area of the conductor, respectively. Any of the three parameters involved in Eq. (5) can be altered by the measurand, thus causing a change of resistance.

Resistive sensors can be classified according to the number of sensing elements that make up the sensor and how these are interconnected, with the following three types:

(a) **Single resistive sensors**, with one sensing element whose resistance (R_x) changes with the measurand, as shown in Fig. 5a. Such a resistance can be modelled as

$$R_x = R_0 \pm \Delta R = R_0(1 \pm x_{\mathrm{R}}) \tag{6}$$

where R_0 is the nominal resistance at a reference value of the measurand, ΔR is the change of resistance due to (and, for some sensors, proportional to) the measurand, and x_{R} is the relative change of resistance (i.e. $x_{\mathrm{R}} = \Delta R/R_0$). These sensors are commonly employed to measure temperature (e.g. platinum sensors and thermistors), light (e.g. light-dependent resistors, LDR), gas (e.g. tin dioxide gas sensors) and humidity.

(b) **Differential resistive sensors**, with two sensing elements (R_{x1} and R_{x2}) that share a terminal, as shown in Fig. 5b, and undergo opposite changes: a change of the measurand causes an increase of R_{x1} and a decrease of R_{x2}, or vice versa. Such resistances can be modelled as

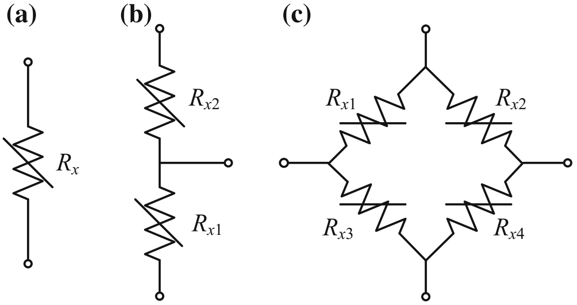


Fig. 5 Resistive sensor with **a** single, **b** differential, and **c** bridge topology

$$\begin{aligned} R_{x1} &= R_0(1 \pm x_R) \\ R_{x2} &= R_0(1 \mp x_R) \end{aligned}, \tag{7}$$

where x_R is assumed to be equal in magnitude but opposite in direction for R_{x1} and R_{x2}. Such a differential topology is quite often implemented through potentiometric sensors that are applied to measure linear or angular position/displacement, pressure (e.g. sensors based on Bourdon tubes) and liquid level (e.g. float-based sensors).

(c) **Bridge-type resistive sensors**, with one, two or four sensing elements in a Wheatstone bridge, thus resulting in a quarter-bridge, half-bridge or full-bridge sensor, respectively. For the full-bridge topology shown in Fig. 5c, which is the most popular since it provides the highest sensitivity, the four sensing elements undergo the same x_R but with opposite signs as follows

$$\begin{aligned} R_{x1} = R_{x4} &= R_0(1 \pm x_R) \\ R_{x2} = R_{x3} &= R_0(1 \mp x_R) \end{aligned}, \tag{8}$$

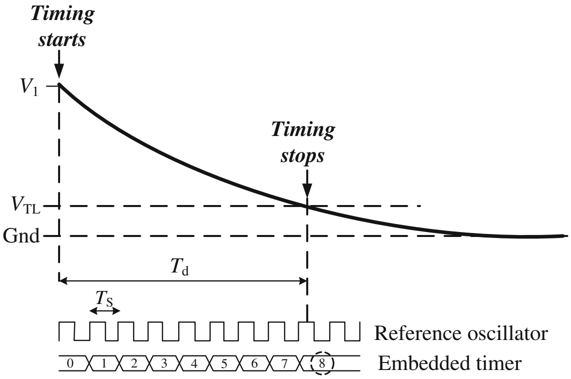
These sensors are commonly used to measure weight (e.g. load cells based on metal strain gages), pressure (e.g. sensors based on semiconductor strain gages) and magnetic field [e.g. Anisotropic (AMR) and Giant (GMR) Magnetoresistive Sensors].

3.2 Microcontroller

A μC is a programmable processor-based digital IC widely used in control and measurement electronic systems. It has three main blocks embedded: (i) a central processing unit (CPU), which executes instructions sequentially; (ii) a memory, which saves the instructions to be executed and data to be processed; and (iii) peripherals, which enable the μC to interact with the off-chip world. The peripherals can be digital (e.g. a timer/counter), analogue (e.g. an analogue comparator), or mixed (e.g. an ADC). However, the direct interface circuits of interest exclusively need digital peripherals, to be precise: input/output digital ports (if possible, with a ST buffer embedded) and a digital timer (if possible, of 16 bits). With regard to the number of bits of the CPU, 8 bits is enough for direct interface circuits, with the corresponding benefits in terms of power consumption.

The tasks of the digital system shown in Fig. 4 can be implemented by a μC following the operating principle represented in Fig. 6. First of all, the start of the discharging-time measurement is synchronized with the timer. Once the measurement has been started, the timer increases by one at every rising edge of its reference oscillator whose period equals T_S. Then, when the exponential discharging voltage crosses the V_{TL} of the ST buffer embedded into Pin 1, the timer stops.

Fig. 6 Discharging-time measurement carried out by the μC



In Fig. 6, the measurement result is the digital number 8, which has information about the value of the sensor resistance included in the RC circuit.

In order to have an accurate measurement of the discharging time shown in Fig. 6, the μC should have the following:

- A crystal oscillator as a reference for the embedded timer, whose temperature coefficient and time drifts are very low.
- A reference oscillator of high frequency (nowadays, it can be up to tens of MHz) to reduce the quantisation error in the discharging-time measurement. The higher the frequency, the better the resolution, but also the higher the power consumption.
- A capture module associated to Pin 1 (see Fig. 4) to automatically capture the value of the timer when the voltage-threshold crossing occurs, regardless of the instruction being executed by the CPU.
- A CPU with a power-down (or sleep) mode to suspend its activity and, hence, to reduce the noise during the discharging-time measurement [16], provided that the timer and the interrupt system keep working in this operating mode. This feature is also of interest to decrease the power consumption.
- An appropriate decoupling capacitor between the power supply pins and a suitable layout of the ground and supply tracks of the PCB to have a clean supply voltage and, consequently, a clean V_{TL} [15].

The measurement of time-modulated signals with a slow slew rate (i.e. a slow transition from ‘1’ to ‘0’, or vice versa) is very susceptible to noise. In the case shown in Fig. 6, the comparison between the two voltages (i.e. the discharging voltage and V_{TL}) can be erroneously triggered due to noise superimposed on either of the two voltages, thus resulting in a wrong value of the digital number. Therefore, any initiative promoting the reduction of trigger noise in the circuit (e.g. power supply noise or CPU-activity noise) will improve the resolution of the measurement.

Nowadays, there are many commercial μCs from different semiconductors companies but with quite similar features that can be employed to build a direct interface circuit. Some examples are: PIC16 family from Microchip Technology, MSP430 family from Texas Instruments, and AVR family from Atmel. Low-power versions of these μCs (e.g. PIC16 with extreme low power technology, or MSP430 with ultra-low power technology) are also available. Direct interface circuits have been implemented using different commercial μCs, but the performance seems to be fairly independent of the μC employed.

3.3 Interface Circuits

The resistive sensor topologies shown in Fig. 5 can be directly measured by a μC through the interface circuits proposed in Fig. 7. In comparison with the circuit shown in Fig. 4, the circuits in Fig. 7 have two additional resistors: R_{i} between Pin 1 and Node 1, which improves the rejection of power supply noise/interference [17] at the expense of a longer charging stage; and R_{s} between Node 1 and the sensor, which ensures that the discharging current is lower than the maximum output current sunk by a port pin even when the sensor resistance is very low.

The direct interface circuit proposed for **single resistive sensors** is shown in Fig. 7a [3], which applies the three-signal auto-calibration technique to have a measurement result insensitive to both multiplicative and additive errors of the circuit [18]. In order to apply such a technique, three measurements are performed sequentially: (1) sensor measurement, which is intended to measure the discharging time through R_x; (2) reference measurement, which is intended to measure the discharging time through a reference resistor (R_{ref}) whose value is known; and (3) offset measurement, which is intended to measure the discharging time through the internal resistance (R_{p}) of the port pins of the μC. The waveform of the voltage across C in a whole measurement is shown in Fig. 8. The state of pins 2, 3 and 4 in Fig. 7a during the discharging stages and the resulting discharging time for each of the three measurements is summarised in Table 1, where $k_{\mathrm{R}} = C \cdot \ln(V_1/V_{\mathrm{TL}})$.

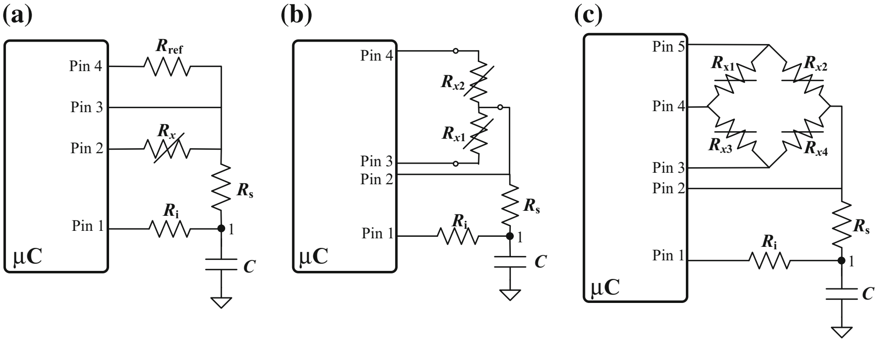


Fig. 7 Interface circuit for a **a** single, **b** differential, and **c** bridge-type resistive sensor

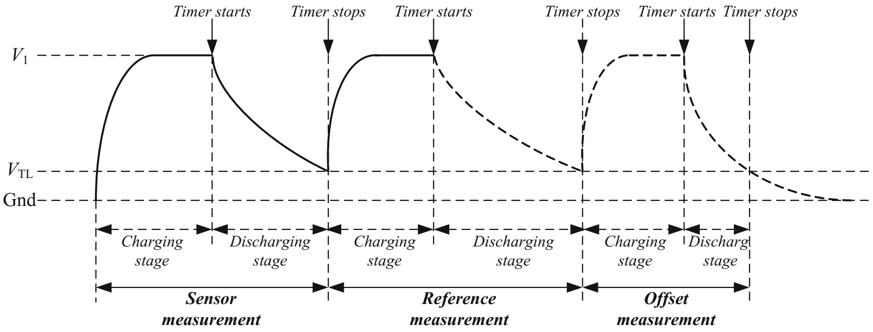


Fig. 8 Waveform of the voltage across C during the charge-discharge process for each of the three measurements involved in the circuit shown in Fig. 7a

Table 1 Pins configuration and discharging times for the circuit in Fig. 7a

Measurement	Pin 2	Pin 3	Pin 4	Discharging time
Sensor	'0'	HZ	HZ	$T_x = k_R(R_s + R_x + R_p)$
Reference	HZ	HZ	'0'	$T_{ref} = k_R(R_s + R_{ref} + R_p)$
Offset	HZ	'0'	HZ	$T_{off} = k_R(R_s + R_p)$

Table 2 Pins configuration and discharging times for the circuit in Fig. 7b

Measurement	Pin 2	Pin 3	Pin 4	Discharging time
Sensor #1	HZ	'0'	HZ	$T_1 = k_R(R_s + R_{x1} + R_p)$
Sensor #2	HZ	HZ	'0'	$T_2 = k_R(R_s + R_{x2} + R_p)$
Offset	'0'	HZ	HZ	$T_{off} = k_R(R_s + R_p)$

Using the three discharging times (T_x, T_{ref} and T_{off}), the sensor resistance can be estimated by

$$R_x^* = \frac{T_x - T_{off}}{T_{ref} - T_{off}} R_{ref}, \tag{9}$$

which is insensitive to the tolerance and low-frequency variability of C, V_1 and V_{TL}. A circuit similar to that shown in Fig. 7a but including diodes and switches has also been proposed to measure remote resistive sensors cancelling the effects of the connecting lead resistances [19].

For **differential resistive sensors**, we propose the direct interface circuit shown in Fig. 7b [4], which also carries out three measurements: (1) sensor measurement #1, (2) sensor measurement #2, and (3) offset measurement, which are intended to measure the discharging time through R_{x1}, R_{x2} and R_p, respectively, applying the pins configuration indicated in Table 2. Using the three discharging times (T_1, T_2 and T_{off}), the parameter x_R of the differential sensor can be estimated by

Table 3 Pins configuration and discharging times for the circuit in Fig. 7c

Measurement	Pin 2	Pin 3	Pin 4	Pin 5	Discharging time
Sensor #1	HZ	'0'	HZ	HZ	$T_1 = k_R[R_s + (R_{x4} \| (R_{x1} + R_{x2} + R_{x3})) + R_p]$
Sensor #2	HZ	HZ	'0'	HZ	$T_2 = k_R[R_s + ((R_{x3} + R_{x4}) \| (R_{x1} + R_{x2})) + R_p]$
Sensor #3	HZ	HZ	HZ	'0'	$T_3 = k_R[R_s + (R_{x2} \| (R_{x1} + R_{x3} + R_{x4})) + R_p]$
Offset	'0'	HZ	HZ	HZ	$T_{off} = k_R(R_s + R_p)$

$$x_R^* = \frac{T_1 - T_2}{T_1 + T_2 - 2T_{off}}. \tag{10}$$

Note that here it is better to estimate the measurand by means of x_R rather than R_{x1} (or R_{x2}), since R_{x1} (or R_{x2}) can also be altered by undesired inputs such as temperature, thus causing multiplicative errors. Moreover, unlike the measurement of single resistive sensors, here x_R can be estimated without using any reference resistor.

Resistive sensors in a **bridge topology** can be directly connected to a μC using the interface circuit shown in Fig. 7c [5, 20]. This circuit measures four discharging times (T_1, T_2, T_3 and T_{off}) applying the pins configuration indicated in Table 3, where the symbol "||" means in parallel. For a full-bridge topology, the parameter x_R of the sensor can be estimated by

$$x_R^* = \frac{T_1 - T_3}{T_2 - T_{off}}. \tag{11}$$

For other bridge topologies, x_R can be estimated using other time-based equations [5]. Furthermore, for sensors whose output is temperature dependent (e.g. piezoresistive pressure sensors), the result obtained from Eq. (11) can be easily corrected by estimating the temperature through the sensor itself [21].

3.4 Uncertainty Sources

The direct interface circuits proposed in Fig. 7 measure three (or four) discharging times and then use them to estimate the sensor resistance or the relative change of resistance through Eqs. (9)–(11). This operating principle involves the following uncertainty sources:

(a) **Mismatch of the internal resistances**: For a CMOS μC, R_p corresponds to the channel resistance of the NMOS transistor embedded into the output buffer that provides a digital ‘0’. In Sect. 3.3, we have assumed that R_p was the same for all the port pins of the µC. However, there is a mismatch between those internal resistances that brings about systematic errors in the measurement. If this mismatch is considered, the estimated value $(R_x^*$ and $x_R^*)$ can be expressed in function of the actual value $(R_x$ and $x_R)$ for the single [3], differential [4] and bridge [20] topology as follows, respectively,

$$R_x^* \approx (R_x + \Delta R_{23})\left(1 - \frac{\Delta R_{43}}{R_{\text{ref}}}\right) \tag{12}$$

$$x_R^* \approx \left(x_R - \frac{\Delta R_{43}}{2R_0}\right)\left(1 + \frac{\Delta R_{23} + \Delta R_{24}}{2R_0}\right) \tag{13}$$

$$x_R^* \approx \left(x_R + \frac{\Delta R_{35}}{R_0}\right)\left(1 + \frac{\Delta R_{24}}{R_0}\right), \tag{14}$$

where $\Delta R_{23} = R_{p2} - R_{p3}$, $\Delta R_{43} = R_{p4} - R_{p3}$, $\Delta R_{24} = R_{p2} - R_{p4}$ and $\Delta R_{35} = R_{p3} - R_{p5.}$ According to Eqs. (12)–(14), we have $R_x^* = R_x$ and $x_R^* = x_R$ only when the internal resistances are matched, otherwise there are offset and gain errors. If the mismatch between internal resistances (ΔR_p) is a few tenths of ohm [3] and the sensor resistance is higher than 1 kΩ, the resulting $\Delta R_p/R$ is very low. Therefore, offset and gain errors due to internal resistances are expected to be in the range of 0.01%.

(b) **Quantisation**: The starting point of the discharging-time measurement is synchronized with the program executed by the µC and, hence, there is no quantisation error at this point. However, the stopping point does suffer from quantisation effects, as shown in Fig. 6. Because of these, the relation between the discharging time to be measured (T_d) and the measurement result (T_q) has a quantisation error (i.e. $T_q - T_d$) that ranges from $-T_S$ to 0 [4]. Equations (9)–(11) can compensate for offset and gain errors obtained during the discharging-time measurements, but not for the non-linearity error caused by quantisation. This non-linearity error in the discharging-time measurement causes offset, gain and/or non-linearity errors in the estimation of R_x and x_R [4]. However, such errors are very low when a high-value capacitance (C in Fig. 7) is employed, but at the expense of a longer measuring time.

(c) **Trigger noise**: The use of a high-value capacitance to reduce the effects of quantisation can cause some non-desired secondary effects. The higher the capacitance, the lower the slew rate at the stopping point of the discharging-time measurement and, hence, the higher the effects of noise coming from the supply voltage or the activity of the CPU. Due to this noise, the discharging-time measurement shows some variability, i.e. different digital numbers are obtained in the digital timer for the same value of the measurand, thus limiting the resolution. Accordingly, the measurement resolution is not only limited by quantisation but also by the trigger noise [22]. Nevertheless and unlike what happens with quantisation effects, trigger noise effects can be reduced by averaging provided that the noise is random, but again at the expense of a longer measuring time.

(d) **Others**: There are other uncertainty sources affecting the measurement but their effects are expected to be less significant. A first example is the *input leakage current* of the pins set as a HZ input during the discharging stage. However, in modern low-power microcontrollers (e.g. PIC16F with extreme low power technology), such a leakage current is 5 nA that is a million times lower than the operating current of the circuit when measuring resistances of units of kiloohm and, therefore, their effects are negligible. Another minor uncertainty source is the *dielectric absorption* (DA) of the capacitor of the RC circuit. To cope with that, it is not advisable the use of electrolytic capacitors whose DA is higher than 10%, but the use of "poly" type capacitors, such as polycarbonate or polypropylene, whose DA is lower than 0.1%.

3.5 Applications

The direct interface circuits for resistive sensors shown in Fig. 7 have been applied to measure many physical and chemical quantities, for example: temperature [3], magnetic field [5], atmospheric pressure [23, 24], gas [25, 26], light [27] and

Table 4 Applications of the direct interface circuits for resistive sensors shown in Fig. 7

Reference	[3]	[4]	[5]
µC	PIC16F873[a]	AVR ATtiny2313[b]	MSP430F123[c]
Supply voltage	5 V	5 V	3 V
Ref. oscillator	20 MHz	20 MHz	4 MHz
Sensor	Temperature sensor (Pt1000)	Potentiometric sensor (1 kΩ)	Magnetoresistive sensor (HMC1052[d])
Topology	Single	Differential	Full-bridge
Interface circuit	Figure 7a	Figure 7b	Figure 7c
Capacitor (C)	2.2 µF	470 nF	2.2 µF
Other components	R_{ref} = 1470 Ω	R_{s} = 470 Ω	R_{i} = 120 Ω
	R_{s} = 330 Ω	R_{i} = 100 Ω	
Meas. range	[−45, 120] °C	[−100, 100]%[e]	[75, 600] µT
Max. NLE[f]	0.01% FSS[g]	0.01% FSS	1.8% FSS
ENOB[h] (measuring time)	11 b (5 ms) or 12.5 b (50 ms)	11.5 b (1 ms) or 13 b (100 ms)	7 b (50 ms)

[a]From Microchip Technology. [b]From Atmel. [c]From Texas Instruments. [d]From Honeywell. [e]Such a range means that the movable common terminal of the potentiometric sensor moves from one end to the other. [f]NLE stands for non-linearity error. [g]FSS stands for Full-Scale Span. [h]ENOB stands for Effective Number Of resolution Bits

respiratory rate [28]. The performance of these circuits in some of the previous applications using different commercial μCs is summarised in Table 4.

Taking into account the simplicity of the proposed interface circuits, the values of non-linearity and resolution shown in Table 4 for the first two cases [3, 4] are quite remarkable. In these cases, the non-linearity error is mainly due to the effects of quantisation in the discharging-time measurement, whereas the resolution is determined by the effects of both quantisation and noise affecting the voltage-threshold crossing. The experimental results for the third case in Table 4 [5], however, are not as excellent as the previous ones. On the one hand, this is due to the non-linearity of the commercial sensor tested; in other words: if the direct interface circuit in Fig. 7c measures a bridge circuit emulated by resistors instead of such a sensor, the maximum non-linearity error of the circuit is about 0.1% FSS. On the other hand, the lower value of resolution is due to the low sensitivity of the commercial sensor. As a rule of thumb, direct interface circuits are able to detect changes of resistance of about 0.1 Ω, which is a very low value when measuring a temperature sensor [3] but not when measuring such a magnetoresistive sensor whose dynamic range is around ±6 Ω.

4 Interfacing Resistive Sensor Arrays to FPGAs

As the complexity of the system and the number of sensors to be measured increase (e.g. array sensors composed by a high number of sensing units), μCs may not have enough resources to implement the techniques described in Sect. 3. In such a case, FPGAs can be a good alternative since they have a high number of I/O pins and also reconfigurable hardware resources to build timer-capture modules operating in parallel. Next, we explain the main features of array sensors and FPGAs, and how to join them to build a direct sensor-to-FPGA interface circuit.

4.1 Array Sensor

Array sensors are composed of many sensing units. These arrays are built to obtain spatial patterns (for instance, a pressure map in tactile sensors) or exploit redundancy to improve sensitivity or selectivity (for example, in electronic noses). Many array sensors are small and implemented with microelectromechanical technologies or conventional technologies for ICs. This is the case of smart vision chips or arrays of thermopiles for infrared imaging. These array sensors commonly incorporate signal conditioning circuitry on the same substrate and, hence, the concept of direct interfacing is not the best choice for them. However, there are discrete arrays of sensors, such as arrays of MOX gas sensors [29], whose interface with the

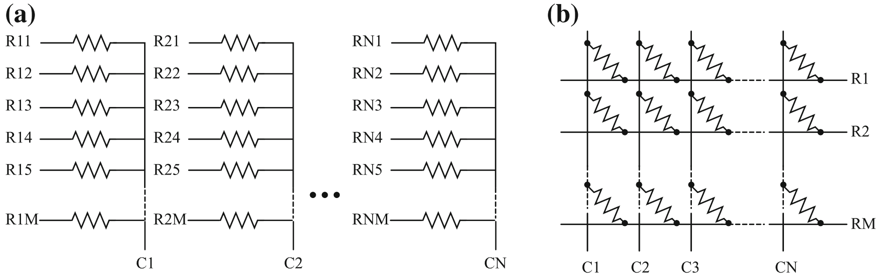


Fig. 9 Architecture of an array sensor **a** with one selection track per sensing unit, and **b** addressed in rows and columns

processing electronics could be noticeably simplified through direct connection. Moreover, direct interfacing is especially suitable for large-size sensors of different shapes, for instance those made with printable electronics. Conductive polymer gas sensor arrays [30], thermal imaging sensors [31] and tactile sensors [32] have been implemented with these technologies.

Regarding its architecture, an array sensor with *M* rows and *N* columns can be built with either one selection track per sensing unit in the array or organized in a row and column fashion where many sensing units share the selection tracks, as shown in Fig. 9a, b, respectively. The latter is obviously advantageous in terms of cost and complexity of hardware. However, shared connections create parasitic current paths that may originate crosstalk between sensing units.

4.2 FPGA

FPGAs are close to ASICs in terms of performance for real-time computing, although ASICs exhibit better dynamic response-power consumption trade-off. The main advantage of FPGAs is that they are programmable, thus allowing rapid system prototyping at low cost. FPGAs are basically composed of cells that have local memory such as flip-flops and are able to perform logic functions. These logic cells are connected through switches to vertical and horizontal routing channels, so the hardware is configurable. The same routing matrix connects the logic cells to a high set of I/O pins. Besides the distributed local memory, FPGAs usually have memory blocks and may incorporate more complex blocks such as multipliers. Advanced versions of FPGA also implement processors as embedded cores, thus resulting in powerful devices called Programmable Systems-on-Chip. The embedded processors can be programmed in high-level languages, while configurable logic is programmed with graphical tools such as circuit schematics or with hardware description languages (HDL). The main vendors of FPGAs are Xilinx and Altera providing a large portfolio of devices with different technologies, number of

I/O pins, number of logic cells, memory and dedicated resources for digital signal processing and communications, and performance in terms of power consumption.

The architecture of an FPGA described before allows replicating the same block (e.g. a digital timer) and, consequently, these devices are capable of parallel signal processing and computation, which is of high interest in digital signal processing and robotics applications. This inherent parallel processing capability increases the bandwidth and reduces the input-output delay in control loops in comparison with the sequential operation of a μC. In addition, since FPGAs are mainly intended to interface to digital devices, they have a high number of I/O pins. These two characteristics make them especially suitable to implement direct interfaces for array sensors. Note, however, that FPGAs are more power demanding than μCs. Moreover, FPGAs are much more limited than μCs regarding the interface to analogue electronics because they do not commonly include ADCs or analogue comparators.

Regarding the issues involved in the concept of direct interfacing depicted in Fig. 4, FPGAs can easily be configured to have embedded digital timers measuring the discharging time of the RC circuit. They also have enough I/O digital pins to control the charge/discharge process of the RC circuit. These I/O pins, however, do not commonly have a ST buffer. In any case, the flexibility of the FPGA allows building dedicated hardware capture modules with similar or better performance than that of ST buffers in terms of noise rejection. I/O drivers are flexible and can be set in different modes such as ‘high impedance’ and ‘strong drive’. The current sunk or sourced by an I/O pin is obviously limited, as in a μC. I/O pins also have non-zero output impedance, which may cause crosstalk errors in the measurement circuit. The clock block in FPGAs allows a flexible managing of the clock, for instance to increase the frequency of the reference clock signal and reduce the quantisation error. Current FPGAs can run at several hundreds of MHz.

Finally, the same requirements as those mentioned in Sect. 3.2 to have an accurate measurement of the discharging time apply here. Specifically, a stable reference oscillator, careful layout design, and decoupling capacitors are necessary.

4.3 Interface Circuits

If the measurement system has a large set of resistive sensors in an array topology, the straightforward approach to measuring them is to use a replica of the circuit in Fig. 7a for each sensing unit. Since the circuits for each sensor would be independent of each other, this approach would not suffer from crosstalk between sensors. However, a number of connections at least as high as two times the number of sensors would be required to address the array sensor. Moreover, a capacitor and a timer-capture module would also be needed per sensor.

A step in the direction of reducing the cost and complexity of the hardware is to share the capacitor and the timer-capture module. This can be done with the architecture of Fig. 9a and the interface circuit shown in Fig. 10. The array sensor is

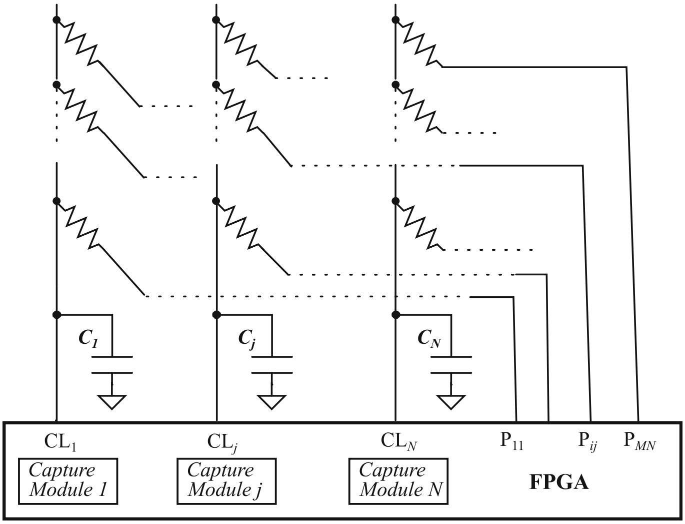


Fig. 10 Interface circuit for an array sensor with one selection track per sensing unit

read as follows. First, the capacitors C_j with $1 \leq j \leq N$ are charged by setting pins CL$_j$ to '1' and the remaining I/O pins to HZ. Then, a whole row is selected by setting its corresponding I/O pins to '0'. For instance, pins of the ith row P$_{ij}$ with $1 \leq j \leq N$ are set to '0' while the remaining pins P$_{kj}$ with $k \neq i$ are set to HZ. The capacitors are then discharged through the sensing resistances of that row and the exponential discharging voltages across them are monitored independently by pins CL$_j$, which are set to HZ. A set of timers are started at the beginning of the discharging phase and they are stopped when V_{TL} is reached at the related column pins. Therefore, a whole row is read in parallel. Techniques similar to those described in Sect. 3.3 can also be applied here to improve the accuracy by adding reference resistors. In this case, a row of reference resistors can be added to carry out the three-signal auto-calibration technique.

Large array sensors addressed in rows and columns, as shown in Fig. 9b, can be interfaced to the FPGA using the circuit shown in Fig. 11, where passive integrators are replaced by active ones implemented by OpAmps. The basics of this circuit are explained through Fig. 12 involving two stages. In the charging stage shown in Fig. 12a, the OpAmp is shut-down by pin Sh and C is charged to V_1. In the discharging stage shown in Fig. 12b, the timer starts, the OpAmp is turned on and the current through R_x is integrated into C. Therefore, the voltage at Pin 1 linearly decreases until V_{TL} is reached and then the timer stops, as shown in Fig. 12c. The discharging time can be expressed as

$$T_d = R_x C(V_1 - V_{TL})/V_1 \tag{15}$$

As in Fig. 10, the array sensor in Fig. 11 is scanned so that all the resistances in a row are measured at the same time. In a first phase, the selection pins P$_i$ with

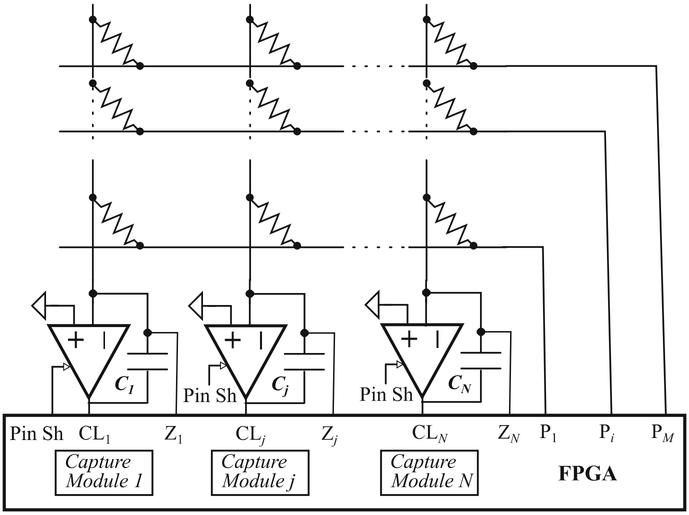


Fig. 11 Interface circuit for an array sensor addressed in rows and columns

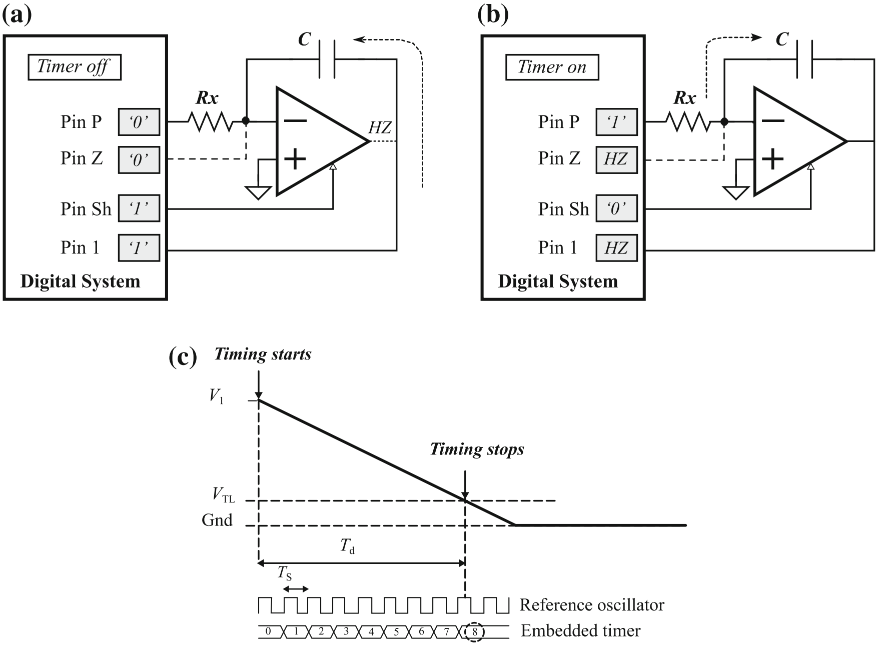


Fig. 12 Basic topology of the interface circuit with an active integrator employed in Fig. 11. Pin configuration during **a** the charging stage, and **b** the discharging stage, and **c** the resulting discharging time

$1 \le i \le M$ are set to ‘0’, CL$_j$ with $1 \le j \le N$ are set to ‘1’, pins Z$_j$ are set to ‘0’, Pin Sh is set to ‘1’ and the capacitors C_j are charged to V_1. In the second phase, a row is selected and the set of timers start counting. For instance, pin P$_i$ is set to ‘1’, thus resulting in a voltage drop V_1 across the resistances R_{ij}. The OpAmps are turned on by setting pin Sh to ‘0’, and pins CL$_j$ and Z$_j$ are now at HZ. Therefore, currents $i_{Dj} = V_1/R_{ij}$ flow into the integrators and, consequently, the voltages at pins CL$_j$ decrease until V_{TL} is reached at every pin CL$_j$, with the corresponding stop of the timer. At this time, Z$_j$ is set to ‘0’ so as to avoid that the voltage at the inverting input of the OpAmp grows and interferes the measurement of other resistances. Note that the columns in Fig. 11 are virtually grounded thanks to the negative feedback loop of the OpAmp, thus following a common strategy [33] to short circuit the non-selected resistances and, hence, to avoid any contributing parasitic current to the output.

The total number of I/O pins dedicated to address the array sensor is $(M + 1) \times N$ in Fig. 10, whereas is $2 \times N + M + 1$ in Fig. 11. For instance, an array of 8 × 8 resistances requires 72 I/O pins in Fig. 10, but 25 in Fig. 11.

4.4 Uncertainty Sources

The interface circuits for array sensors shown in Figs. 10 and 11 suffer from uncertainty sources similar to those described in Sect. 3.4. However, additional errors arise due to the I/O features of the FPGA and to the array nature of the sensors. These novel uncertainty sources are described next.

(a) **Trigger noise**: As said in Sect. 3.4, trigger noise alters the threshold voltage and the discharging voltage signal, so the discharging time is affected in consequence. The use of an I/O pin with a ST buffer in µCs reduces this uncertainty because crosses of the discharging signal with the threshold after the first one are ignored thanks to the hysteresis of the buffer. Unfortunately, FPGAs generally do not have ST input buffers so the contribution of the trigger noise is significant in a straightforward realization where the output of the input buffer is used to stop the timer. Embedded resources of the FPGA can be configured to build smart capture modules [34] that detect the first change of logical value at the input buffer when the input signal reaches the threshold (label F in Fig. 13). This can be done by adding positive feedback in digital circuits to achieve the memory of the hysteresis cycle or with a level triggered latch. In addition, the flexibility of the storage elements in the FPGA to be synchronized with both edges of the clock signal, and also the detection of not only the first (label F in Fig. 13) but the last (label L in Fig. 13) transition at the output of the input buffer can be exploited to carry out averaging. This actually filters part of the trigger noise and achieves more precision without losing bandwidth.

(b) **Crosstalk**: An additional source of uncertainty in the sensor topologies shown in Fig. 9 with respect to those described in Sect. 3 is crosstalk. The sharing of circuit

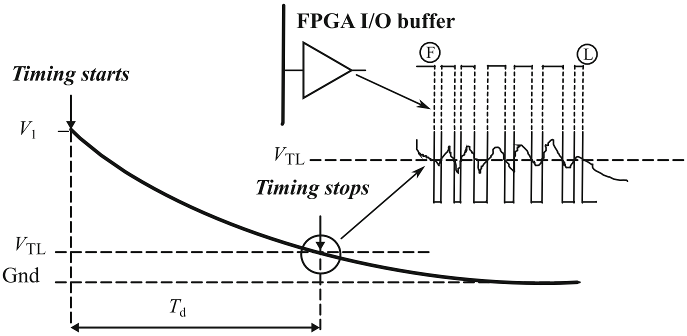


Fig. 13 Effects of trigger noise when a ST buffer is not employed

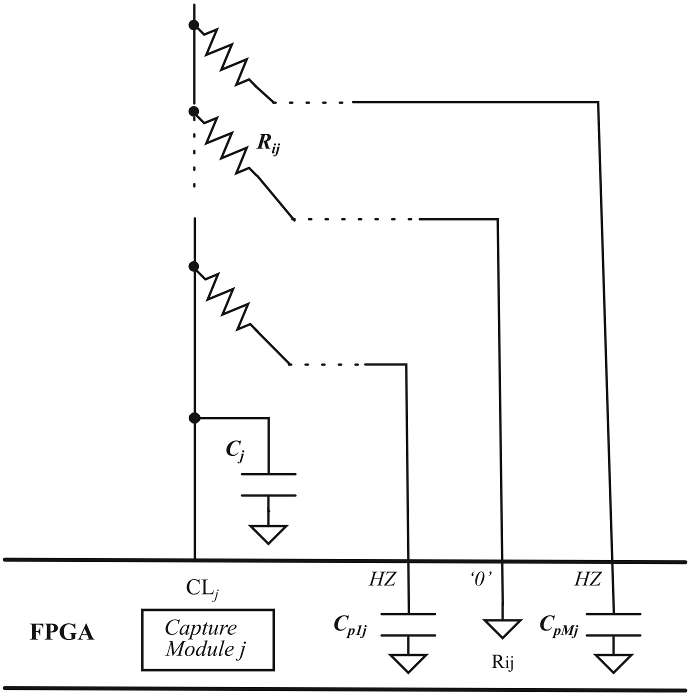


Fig. 14 Crosstalk due to the parasitic capacitances in the interface circuit shown in Fig. 10

components to lower the cost has the drawback of introducing such an error. As a consequence, the timing does not depend only on the value of the resistance that is being measured but also on the value of other resistances in the array.

Regarding the interface circuit in Fig. 10, crosstalk is mainly due to parasitic capacitances associated to connection tracks and I/O buffers set at HZ. If they are taken into account, the discharging circuit is not that in Fig. 4c but the one depicted in Fig. 14 for the column j, where C_{pkj} with $1 \leq k \leq M$ and $k \neq i$ are such parasitic capacitances. For a given R_{ij}, the higher the resistance of the non-selected rows, the shorter the discharging time T_{dij}, while its maximum value will be registered when all these resistances are minimum. Therefore, a higher range of resistances increases the difference between the minimum and maximum values of T_{dij} due to crosstalk and the uncertainty in T_{dij} in consequence. A worst case estimation can be done for a maximum range. In this case, for $R_{kj} \to \infty$ with $1 \leq k \leq M$ and $k \neq i$, T_{dij} is minimum and takes the value

$$T_{dij(min)} = R_{ij} C_j \ln\left(\frac{V_1}{V_{\mathrm{TL}j}}\right). \tag{16}$$

On the other hand, for $R_{kj} \to 0$, T_{dij} is maximum and results in

$$T_{dij(max)} = R_{ij} C_{eqij} \ln\left(\frac{V_1}{V_{\mathrm{TL}j}}\right), \tag{17}$$

where $C_{eqij} = C_j + \sum_{\substack{k=1 \\ k \neq i}}^{M} C_{pkj}$. Therefore, $T_{dij(min)} \leq T_{dij} \leq T_{dij(max)}$ and its actual value depends on the specific values of the remaining resistances in that column. For a uniform distribution of these resistances, the relative standard uncertainty of T_{dij} generated by crosstalk is given by

$$\frac{u(T_{dij})}{T_{dij}} = \frac{\sum\limits_{\substack{k=1 \\ k \neq i}}^{M} C_{pkj}}{\sqrt{12} C_j}, \tag{18}$$

From (18), the higher the aggregated parasitic capacitance with respect to C_j, the higher the relative uncertainty. Such an aggregated parasitic capacitance increases with M and depends on the circuit layout and technology used to implement it. The relative uncertainty in (18) can be reduced with a higher value of C_j, although there is a tradeoff with the measuring time and with the trigger noise since both increase with C_j. On the other hand, a high value of C_j reduces the uncertainty due to quantisation. The aggregation of the errors caused by all these sources determines the resolution of the measurement.

The circuit in Fig. 11 is subjected to more crosstalk error sources than that in Fig. 10. The three main error sources are: (i) the internal resistance of the output buffer that sources current to the selected row (i.e. R_{pi} in Fig. 15), (ii) the internal resistance of the output buffer that sinks current from the column once the discharging process has ended (i.e. R_{nj} in Fig. 15), (iii) the input offset voltage of the OpAmps (i.e. V_{off} in Fig. 15). Because of these, the discharging time does not

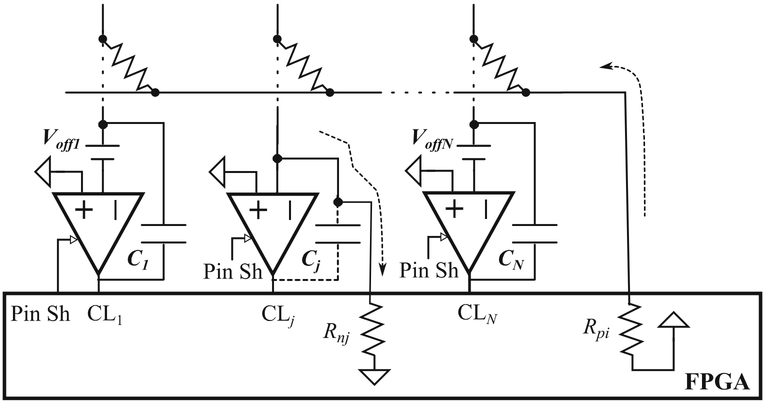


Fig. 15 Crosstalk due to the I/O driver impedance and the input offset voltage of the OpAmp in the interface circuit shown in Fig. 11

depend only on the resistance being measured but also on other resistances in the array.

As for the effects of R_{pi} in Fig. 15, the discharging time in (15) is modified as

$$T_{dij} = R_{ij} C_j \frac{V_1 - V_{TLj}}{V_1} \frac{R_{pi} + R_{eqi}}{R_{eqi}}, \tag{19}$$

where R_{eqi} is the equivalent parallel resistance of the resistances in the row selected. Note from (19) that the lower R_{eqi} with respect to R_{pi}, the higher the crosstalk. This imposes a lower limit of the range of resistances and also limits the number of columns in the array for a given accuracy. The addition of known reference resistors in a new reference column is proposed in [35] to obtain the following expression

$$T_{dij} = \frac{R_{ij}}{R_{ic}} \frac{C_j}{C_c} \frac{V_1 - V_{TLj}}{V_1 - V_{TLc}} T_{dic}, \tag{20}$$

where R_{ic}, C_c, V_{TLc} and T_{dic} are the row resistance, the capacitor, the input buffer threshold voltage and the discharging time associated to the reference column, respectively. Since the other resistances in the array are not in (20), the crosstalk caused by R_{pi} is cancelled. Moreover, the use of known reference resistors in another new reference row provides an expression of the discharging time that only depends on the known value of the reference resistors and the measured discharging times associated to these resistors and R_{ij} [35].

The crosstalk caused by R_{nj} in Fig. 15 is related to the role of pin Z to clamp the voltage at the column to zero. As said in Sect. 4.3, since non-selected rows are also driven by a zero voltage signal, parasitic resistive paths are in principle short

Table 5 Applications of the interface circuits shown in Figs. 10 and 11

Reference	[37]	[35]
FPGA	Spartan3AN[a]	Spartan3AN[a]
OpAmp	–	TLV2475N[b]
Supply voltage	3.3 V	3.3 V
Ref. oscillator	50 MHz	50 MHz
Technology	PCB, insertion sockets, axial-lead discrete resistors	
Sensor topology	Figure 9a	Figure 9b
Sensor size	M = 7, N = 8	M = 8, N = 6
Interface circuit	Figure 10	Figure 11
Capacitor (C)	47 nF	47 nF
Meas. range	[200, 7350] Ω	[556, 3159] Ω[c]/[3296, 9975] Ω[d]
ENOB (scan time)[e]	12.2 b (5 ms)	7.9/8.6 b (5 ms)
Max. NLE	0.028% FSS	0.038% FSS/0.037% FSS

[a]From Xilinx. [b]From Texas Instruments. [c]The OpAmp output does not saturate, i.e. Eq. (21) is satisfied. [d]The voltage at column jth is clamped to zero voltage with pin Z_j when the threshold is reached. [e]Scan time is the time to read the whole array

circuited and then crosstalk is reduced. However, if the sensor resistance to be measured is low, the current sunk by Pin Z is high, thus generating a voltage rise at the column that causes parasitic currents and then crosstalk errors. This error can be neglected for high enough values of the sensor resistance (see Table 5 in Sect. 4.5). A minimum resistance is also required to accomplish with the maximum current that is able to source the pin that selects the row and to sink the pin Z. A direct strategy to overcome this limitation and reduce the crosstalk errors consists in adding external resistors in series with R_{pi} to limit the current. In this case, note that Eq. (20) is still valid since the measurements obtained from the reference column provide indirect estimations of these resistances.

Another alternative to reduce the crosstalk caused by R_{nj} consists in ensuring that OpAmps always work in the linear region so the negative feedback imposes always a voltage close to ground. This is achieved if the range of resistances values is set to guarantee that the longest discharging time corresponding to the highest resistance is short enough to avoid that the output of the OpAmp of the column that reads the smallest resistance (i.e. shortest discharging time) saturates. This is achieved if

$$R_L \leq R_{ij} \leq \frac{V_1}{V_1 - V_{Tj}} R_L, \tag{21}$$

where R_L is the lowest resistance value in the array. A strategy to increase the range imposed by (21) is the reading of two rows at the same time, the one being scanned and a reference row with known resistances. In this way, the reference resistors are in parallel with the ones being read and the equivalent maximum resistance is lowered, so Eq. (21) is met by the equivalent parallel resistances but the actual range of the resistances in the array is much higher [35].

The input offset voltage of the OpAmps in Fig. 15 is also a source of crosstalk error because it changes the voltage at the corresponding column with respect to ground, thus causing parasitic currents in the array. Input bias currents of the OpAmps also introduce error since they are added to the current being integrated and change the discharging time. A procedure is proposed in [35] to reduce these second-order effects where two reference rows and one reference column with known resistances are added to the array. The target resistance R_{ij} can be expressed as a function of the known resistors and the associated discharging times only, and the effects of the offset voltages and bias currents are cancelled.

4.5 Applications

The interface circuits presented before have been applied to the measurement of tactile array sensors in [36] using an FPGA as a digital system. In addition, [37] and [35] report results for the interface circuits shown in Figs. 10 and 11 applied to the measurement of a generic array sensor made of discrete-lead axial resistors in insertion sockets on a PCB. Table 5 shows a summary of the results for different resistive ranges. Note that the time to read the whole array is 5 ms in all cases so a very fast scanning is achieved thanks to the parallel acquisition. Moreover, a resolution as high as 12.2 ENOB for the circuit in Fig. 10 was experimentally measured for two times the standard deviation as error estimation and 500 samples. The circuit in Fig. 11 requires much less I/O pins to address the array at the expense of more complex circuitry with more uncertainty sources. A resolution around 8 bits was reached, although it can increase for higher resistance values and wider measurement ranges.

5 Conclusions

After explaining such circuits and techniques for the direct connection of resistive sensors to digital systems, the following conclusions can be drawn:

- Direct interface circuits clearly simplify the measurement chain because neither an amplifier nor an ADC are needed between the sensor and the digital system. The key element is a digital timer that measures the charging/discharging time of an RC circuit formed by the resistive sensor and a known capacitor.
- A common low-cost general-purpose 8-bit μC can be the core of these sensor interfaces without requiring any on-chip ADC, OpAmp or analogue comparator. For resistive sensor arrays, the use of an FPGA is more advisable because different resistances of the array can be measured in parallel through a set of timers running simultaneously.

- In spite of their simplicity and low cost, these sensor interfaces have a satisfactory performance in terms of linearity and resolution and, therefore, they are very attractive for medium-accuracy, medium-resolution applications. A measuring time of around units or tens of millisecond can be their main limitation if the quantity to be measured changes quite fast.
- Since very similar results have been obtained when using different commercial digital systems from different manufacturers, the design of these sensor interfaces can be considered independent of any specific device or IC from any manufacturer.

From the authors' point of view, future research work on direct interface circuits could be focussed on the following directions:

- The analysis of the limitations when measuring resistive or capacitive sensors subjected to dynamic changes, and not quasi-static changes as considered so far.
- The direct measurement of other types of sensor. For instance, the use of digital timers to directly measure sensors providing an amplitude-modulated sinusoidal voltage signal.
- The use of new digital peripherals, such as configurable logic cells, embedded into the new generation of µCs to improve the performance of the proposed sensor interfaces and/or to develop novel operating principles.

Acknowledgements This work has been partially funded by the Spanish Government and by the European ERDF program funds under contract TEC2015-67642-R.

References

1. R. Pallàs-Areny, J.G. Webster *Sensors and Signal Conditioning*, 2nd edn. (Wiley, New York, 2001)
2. J.H. Huising, Smart sensor systems: Why? Where? How?, in *Smart Sensor Systems*, ed. by G.C.M. Meijer (Wiley, Chichester (UK), 2008), pp. 1–21
3. F. Reverter, J. Jordana, M. Gasulla, R. Pallàs-Areny, Accuracy and resolution of direct resistive sensor-to-microcontroller interfaces. Sens. Actuators, A **121**, 78–87 (2005)
4. F. Reverter, O. Casas, Interfacing differential resistive sensors to microcontrollers: a direct approach. IEEE Trans. Instrum. Meas. **58**(10), 3405–3410 (2009)
5. E. Sifuentes, O. Casas, F. Reverter, R. Pallàs-Areny, Direct interface circuit to linearise resistive sensor bridges. Sens. Actuators, A **147**, 210–215 (2008)
6. F. Reverter, O. Casas, Direct interface circuit for capacitive humidity sensors. Sens. Actuators, A **143**, 315–322 (2008)
7. F. Reverter, O. Casas, Interfacing differential capacitive sensors to microcontrollers: a direct approach. IEEE Trans. Instrum. Meas. **59**(10), 2763–2769 (2010)
8. F. Reverter, O. Casas, A microcontroller-based interface circuit for lossy capacitive sensors. Meas. Sci. Technol. **21** (065203), 8 pp (2010)
9. Z. Kokolanski, J. Jordana, M. Gasulla, V. Dimcev, F. Reverter, Direct inductive sensor-to-microcontroller interface circuit. Sens. Actuators, A **224**, 185–191 (2015)
10. N. Ramadoss, B. George, A simple microcontroller based digitizer for differential inductive sensors, in *Proceedings IEEE I2MTC*, 11–14 May 2015, Pisa, Italy (2015), pp. 148–153

11. L. Bengtsson, Direct analog-to-microcontroller interfacing. Sens. Actuators A **179**, 105–113 (2012)
12. F. Reverter, R. Pallàs-Areny, *Direct Sensor-to-Microcontroller Interface Circuits. Design and Characterization* (Marcombo, Barcelona, Spain, 2005)
13. F. Reverter, The art of directly interfacing sensors to microcontrollers. J. Low Power Electron. Appl. **2**, 265–281 (2012)
14. F. Reverter, Power consumption in direct interface circuits. IEEE Trans. Instrum. Meas. **62**(2), 503–509 (2013)
15. F. Reverter, J. Jordana, R. Pallàs-Areny, Internal trigger errors in microcontroller-based measurements, in *Proceedings XVII IMEKO World Congress*, 22–27 June 2003, Dubrovnik, Croatia (2003), pp. 655–658
16. F. Reverter, R. Pallàs-Areny, Uncertainty reduction techniques in microcontroller-based time measurements. Sens. Actuators, A **127**, 74–79 (2006)
17. F. Reverter, M. Gasulla, R. Pallàs-Areny, Analysis of power-supply interference effects on direct sensor-to-microcontroller interfaces. IEEE Trans. Instrum. Meas. **56**(1), 171–177 (2007)
18. G.C.M. Meijer, Interface electronics and measurement techniques for smart sensor systems, in *Smart Sensor Systems*, ed. by G.C.M. Meijer (Wiley, Chichester (UK), 2008), pp. 23–54
19. R.N. Ponnalagu, B. George, J. Kumar, A microcontroller sensor interface suitable for resistive sensors with large lead resistance, in *Proceedings of the International Conference on Sensing Technology*, 2–4 Sept 2014, Liverpool, UK (2014), pp. 327–331
20. E. Sifuentes, O. Casas, F. Reverter, R. Pallàs-Areny, Improved direct interface circuit for resistive full- and half-bridge sensors, in *Proceedings ICEEE 2007*, 5–7 Sept 2007, Mexico City, Mexico (2007), pp. 197–200
21. F. Reverter, G. Horak, V. Bilas, M. Gasulla, Novel and low-cost temperature compensation technique for piezoresistive pressure sensors, in *Proceedings XIX IMEKO World Congress*, 6–11 Sept 2009, Lisbon, Portugal (2009), pp. 2084–2087
22. F. Reverter, R. Pallàs-Areny, Effective number of resolution bits in direct sensor-to-microcontroller interfaces. Meas. Sci. Technol. **15**, 2157–2162 (2004)
23. J. Jordana, R. Pallàs-Areny, A simple, efficient interface circuit for piezoresistive pressure sensors. Sens. Actuators, A **127**, 69–73 (2006)
24. S.Y. Yurish, A simple and universal resistive-bridge sensors interface. Sens. Transducers J. **10**, 46–59 (2011)
25. J. Courbat, D. Briand, L. Yue, S. Raible, N.F. Rooij, Drop-coated metal-oxide gas sensor on polyimide foil with reduced power consumption for wireless applications. Sens. Actuators, B **161**, 862–868 (2012)
26. S. Knobelspies, B. Bierer, A. Ortiz Perez, J. Wöllenstein, J. Kneer, S. Palzer, Low-cost gas sensing system for the reliable and precise measurement of methane, carbon dioxide and hydrogen sulfide in natural gas and biomethane. Sens. Actuators B **236**, 885–892 (2016)
27. E. Sifuentes, Sensor autónomo para detectar vehículos estáticos. Ph.D. thesis, Universitat Politècnica de Catalunya, Spain (2009) (in Spanish)
28. E. Sifuentes, J. Cota-Ruiz, R. González-Landaeta, Respiratory rate detection by a time-based measurement system. Revista Mexicana de Ingeniería Biomédica **37**(2), 91–99 (2016)
29. L.C. Wang, K.T. Tang, C.T. Kuo, C.L. Ho, S.R. Lin, Y. Sung, C.P. Chang, Portable electronic nose system with chemiresistor sensors to detect and distinguish chemical warfare agents. J. Micro/Nanolith. MEMS MOEMS **9**(3), 302–306 (2010)
30. M. Camara, F. Molina-Lopez, E. Danesh, G. Mattana, A. Bontempi, D. Teyssieux, L. Thiery, P. Breuil, C. Pijolat, K. Persaud, D. Briand, N.F. de Rooij, Printed micro-hotplates on flexible substrates for gas sensing, in *Proceedings TRANSDUCERS'13*, 16–20 July 2013, Barcelona, Spain (2013), pp. 1059–1062
31. http://www.pstsensors.com
32. http://www.tekscan.com
33. T. D'Alessio, Measurement errors in the scanning of piezoresistive sensors arrays. Sens. Actuators, A **72**, 71–76 (1999)

34. Ó. Oballe-Peinado, F. Vidal-Verdú, J.A. Sánchez-Durán, J. Castellanos-Ramos, J.A. Hidalgo-López, Smart capture modules for direct sensor-to-FPGA interfaces. Sensors **15** (12), 31762–31780 (2015)
35. Ó. Oballe-Peinado, F. Vidal-Verdú, J.A. Sánchez-Durán, J. Castellanos-Ramos, J.A. Hidalgo-López, Improved circuits with capacitive feedback for readout resistive sensor arrays. Sensors **16**(2), 149 (2016)
36. F. Vidal-Verdú, Ó. Oballe-Peinado, J.A. Sánchez-Durán, J. Castellanos-Ramos, R. Navas-González, Three realizations and comparison of hardware for piezoresistive tactile sensors. Sensors **11**(3), 3249–3266 (2011)
37. Ó. Oballe-Peinado, F. Vidal-Verdú, J.A. Sánchez-Durán, J. Castellanos-Ramos, J.A. Hidalgo-López, Accuracy and resolution analysis of a direct resistive sensor array to FPGA interface. Sensors **16**(2), 181 (2016)