Digital Sensor based on Timer for Embedded Systems

Letícia Guimarães[1],Celso Maciel[1], Alleff Dymytry P. de Deus[1],Gabrieli M. de Melo[1]

[1] Computer Engineering - Universidade Estadual do Rio Grande do Sul (UERGS)
Santa Maria, 2300 – Bom Fim Velho – Guaíba – RS – Brazil

{leticia-guimaraes@uergs.edu.br, celso-costa@uergs.edu.br, alef-deus@uergs.edu.br, gabrieli-melo@uergs.edu.br}

Abstract. *Embedded instrumentation usually has analog sensors relating to a physical measurement. This analog signal is converted to a digital signal to be connected to a microprocessor. However most of microprocessors presents a built-in analog to digital converters, some of them do not. In this work, we analyze a digital sensor based on a timer circuit providing a pulse width which is proportional to a physical quantity to be measured. The pulse width is proportional to the resistive and capacitive elements, where one of them is the sensor. The experimental results showed a simple circuit as a Do It Yourself –DIY solution for an unavailability of an AD-Converter on embedded instrumentation applications.*

1. Introduction

Educational microprocessor kits are generally a low cost, small size computer, with capable device that enables ordinary people to explore computing. Besides, the user can dispose sensors and actuators to develop instrumentation, automation or entertainment projects. These development kits are available to students for robotics, computer vision, data logger etc. Most of them work with sensors assembled to an appropriate circuit called shields, available on market. The cost of the shields depending on the specs of development kit. Sensor shields for development kits without analog to digital converter are more expensive than shields for kits with ADC embedded. In order to provide a low cost sensor shields we propose a system as shown in Figure 1.

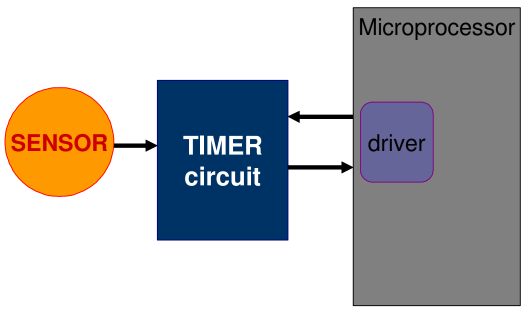

Figure 1 - Scheme of a digital sensor with a timer circuit.

The system is very simple, a timer circuit with a sensor element that causes a change on the width of its output pulse, Figure 1. The pulse has its time in on state counted by a device driver running on the microprocessor. The device driver should communicate with the timer circuit and be set according to the sensor features (resistance or capacitance) and the measurement range. The range adjustment allows performing its best resolution. The use of a timer as a mean to convert a sample to be input in microprocessor has been proposed in many references [Cady 1997, Blick, B. 2002 and Undergroundworkbench 2009]. Despite this alternative has been largely posted, few of them presents an analysis deeper for its application. The timer output is connected to the microprocessor I/O pin. In order to verify the circuit performance some experiments were performed on two circuits: one, assembled with a thermistor as temperature sensor, and another using a potentiometer as angle sensor. The results shown that it is a very simple solution to an unavailability of a traditional AD Converter. Sampling with timer circuits presents the following advantages: Timer circuits are available in the market, as the well-known integrated circuit 555 [ARDUINO 2018]; Have low cost compared to a conventional AD converter; It is a DIY circuit (Do It Yourself circuit) that's very useful when someone need to build a prototype in a short time, specially to students projects.

2. Microprocessor Kits

There are several companies that manufacture kits that provide shields for diverse microprocessor kits with many applications. We present one of the brands as example, since all have the similar features. The Grove™ is a sensor kit developed by Seeed Technology Co. Ltd, with a standard plug provided by the company, and with its drivers available on the its Internet site. The development kit Arduino™ is around US$ 50,00 and the kit for Raspberry Pi TM is around US$ 90,00. These values are not affordable for some students or developers. To minimize the cost the Arduino kit is used as an interface between sensor and Raspberry Pi[4]. This option is advantageous for three main reasons: 1) availability of Arduino [5] kits, 2) the sensors that used are compatible with Arduino and; 3) Arduino can communicate with Raspberry via USB. However, the use of Arduino as intermediate is not practical; the use of two kits for one work is not efficient when we have many students working at the same time.

3. Methodology

3.1. Hardware

The proposed circuit is based on a timer that produces a pulse which the on timing depending on the variation in the value of a resistance or capacitance. There is in the market an integrated circuit design to be a timer, the IC555 [6], Figure 2. Timers built with IC555 produce a pulse signal whose timing varies proportionally to the variation of the measured physical quantity. The IC555 is considered one of the most used ICs during almost 50 years of its launch in the market, and has been included in the most varied projects since then [2]. Figure 3 show a schema of a circuit using an IC555, where the Rsensor could be a sensor as a thermistor (a resistor whose resistance is dependent on temperature), LDR (Light Dependent Resistor), or a potentiometer (angle sensor), for example. In Figure 2 the trigger signal is used as start sampling command. The capacitor C value is based on sampling frequency (fs) and maximum resistive value of the sensor (RM), in order to get the largest pulse as output (TsampleM) consequently the best resolution on conversion, described in (1) and (2). The elements Ro and diode zener are voltage regulator for the microprocessor digital input when its voltage level is lower than supply voltage (+V) of the IC555, see Figure 3.

$$\mathrm{TsampleM} = 1.1 \times \mathrm{RM} \times \mathrm{C} \quad (1)$$

$$fs = \frac{1}{TsampleM} \quad (2)$$

where TsampleM is the maximum available time for digital conversion.

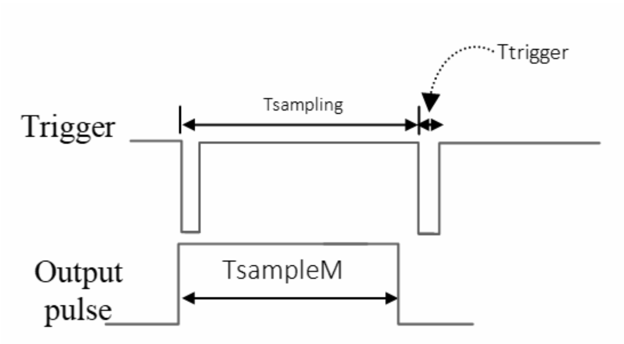

Figure 2 - Timing signal diagram.

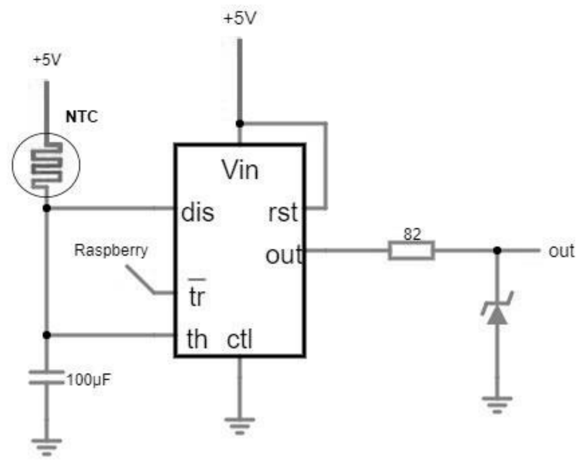

Figure 3 - Temperature sensor based on timer circuit based on IC555.

3.2. Software

The device driver sends the trigger signal to the timer, Figure 2 shows the signaling to start a conversion, then reads the timer output and count the time while the pulse is on. Figure 4 show the flow chart for the sampling driver. The device driver should receive the following parameters: (1) two microprocessor I/O pins, one for trigger pulse output

PO, and other for timer pulse input PI; (2) time of the trigger pulse TTRIG. The pulse width is related to the counter value COUNT.

4. Experiment Results

The experiments proceeded with temperature acquisition. All Measurements were made with twenty samples to take an average as the more probable measured value. Figure 5 shows the flow chart for the program using a timer sensor to acquire twenty samples and take the average value. The average value is taken as the measured value, minimizing the error caused by the jitter . The sample driver is in Figure 4. The program runs in a Raspberry pi written on python language.

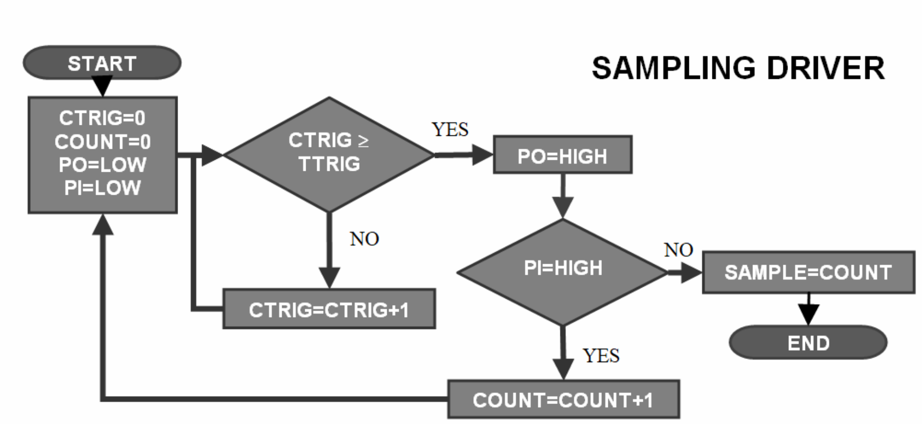

Figure 4 - Flow chart for the sample driver. CTRIG is the trigger time counter, COUNT is the timer pulse width counter, PO is the Trigger signal port, PI is the timer pulse port. TTRIG is the width of the trigger pulse.

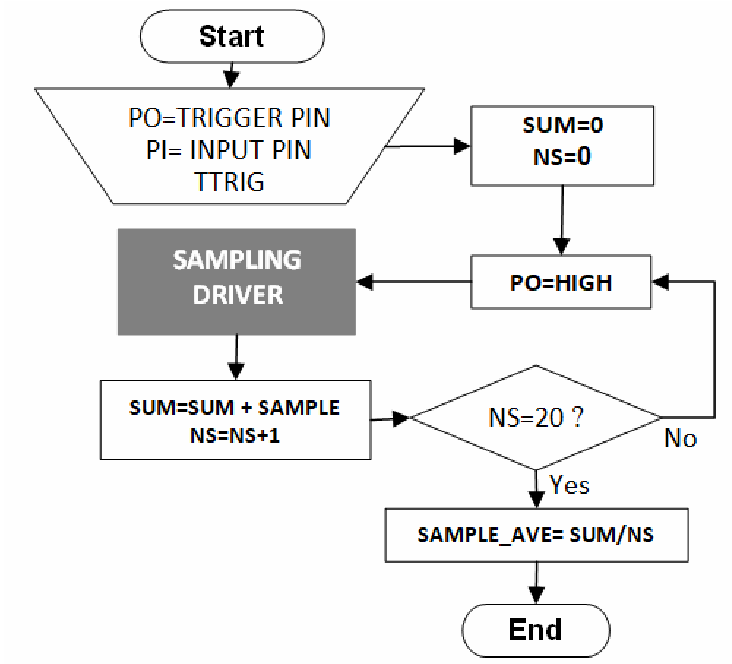

Figure 5 - Flow chart for a program using a timer sensor to acquire twenty samples and take the average value from these twenty samples. The sample driver is in Figure 4.

4.1. Temperature Sensor

The temperature sensor used was a thermistor and IC555 as the timer, see Figure 3 and Figure 6. A thermistor is a thermoresistor with its resistance value change as its temperature, in (3).

$$R(T) = R(T_0) \times e^{\left\{\beta \times \left[\left(1/T\right) - \left(1/T_0\right)\right]\right\}} \tag{3}$$

Where:

R (T) and R (T0) are the thermistor resistance value at temperature T and T_0 (measured in Kelvin), respectively;

e is the neper number or Euler number;

T and T_0 are the actual and a reference temperature, respectively;

β is the thermistor constant for the temperature equation of the NTC (4).

$$\beta = \frac{\ln\left(\frac{R_{T1}}{R_{T2}}\right)}{\frac{1}{T_1} - \frac{1}{T_2}} \tag{4}$$

Where:

R_{T1} and R_{T2} are the resistance values of the thermistor at temperature T1 and T2, respectively.

The thermistor used in our experiments was NTC203, its resistance value changes as its temperature; it is described in (5)

$$R(T) = 20 \times 10^3 \times e^{\left\{\left[3976 \times \left(1/T\right)\right] - \left[1/(25 + 273)\right]\right\}} \tag{5}$$

The capacitor C used was 100nF, the results are shown in Fig.8.

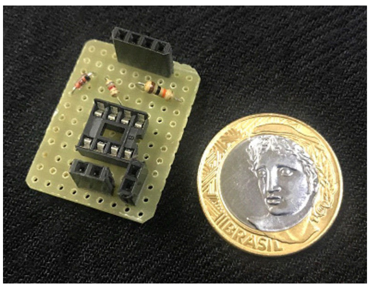

Figure 6 – DIY Timer with IC555 mounted on 2.5cmx3.0cm circuit board.

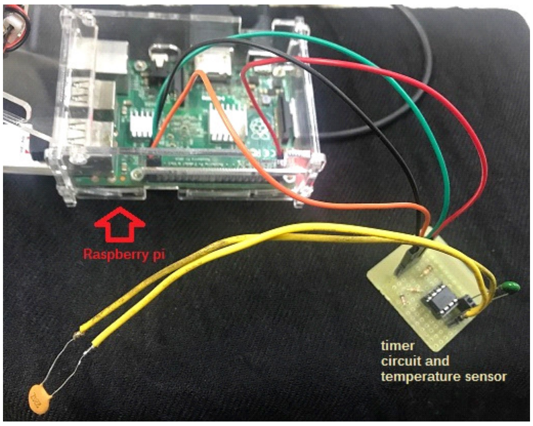

Figure 7 - The temperature sensor connected to a Raspberry Pi.

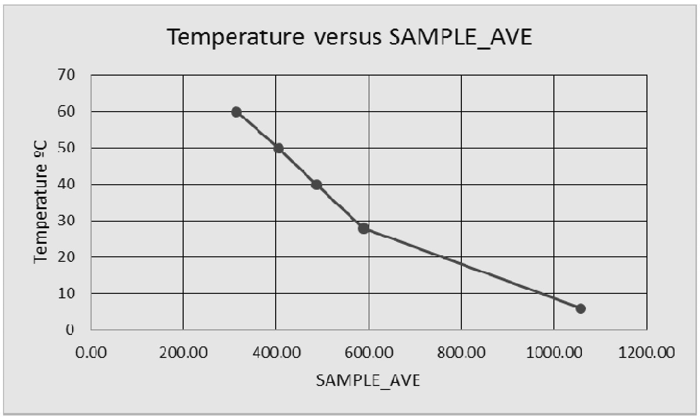

Figure 8 - Temperature versus SAMPLE_AVE.

The results on graphs in Figure 6 and Figure 7 show the behavior of sensor and the sample system, respectively. The values in fig. 6 were calculated using the equation (5). In Figure 7, the driver counter results show the same behavior of the sensor shown in Figure 6. Notice that are at least two lines, defined by equations (7) and (8). While (7) works for the range 0ºC to 28ºC and (8) works for the range 28ºC to 60ºC.

$$\text{Temperature} = (0.0469 \times \text{SAMPLE\_AVE} + 55.63) \tag{7}$$

$$\text{Temperature} = (0.1158 \times \text{SAMPLE\_AVE} + 96.49) \tag{8}$$

4.1.1. Temperature Measurement Specifications

Conversion time and Bandwidth limit: is the time required to complete a conversion of the input signal it is calculated by (1). It establishes the upper signal frequency limit that without aliasing[1]. The Bandwidth limited by the maximum resistance of the sensor and the capacitor value.

Resolution: The resolution is limited by the sampling cycle performed by the sampling driver, the sensor behavior and capacitor used (2). Using the NTC203 the maximum resolution obtained on the experiment was approximately 5×10-3 ºC.

Linearity; There are two linear equations one for each range: one from 0°C to 28ºC and other from 28ºC to 60ºC, described by (7) and (8) respectively. In both experiments the linearity is lower than 1%.

4.2. Angle Position Sensor

The angle position sensor is a potentiometer and IC555 as the timer, see Figure 10. A potentiometer 100Ω has its resistance value related to an angle. The graph in Figure 9 shows the angle position sensor response. Approximating to a linear equation we obtained the equation (9).

$$\text{Angle Position} = 0.11121\times \text{SAMPLE\_AVE} + 13.035 \tag{9}$$

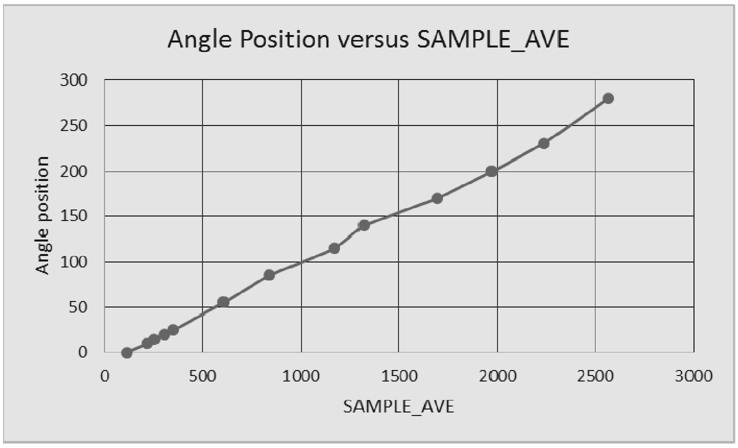

Figure 9 - Angle position versus SAMPLE_AVE.

4.2.1. Angle Position Measurement Specifications

Conversion time and Bandwidth limit: The Bandwidth limited by the maximum resistance of the sensor and the capacitor value, (Rpot=100Ω and C=350nF), fsample until 25,97kHz (1).

Resolution: The resolution is limited by the sampling cycle performed by the sampling driver, the sensor behavior and capacitor used (2). The maximum resolution obtained on the experiment was approximately 0.1 °.

Linearity; There is a linear equation described by (9). In both experiments the linearity is lower than 3%.

5. Conclusions

The results shown that a timer is a very simple solution to an unavailability of a traditional AD Converter. The measurement range can be optimized to the application according to the resistive sensor parameters and by choosing the suitable value for the capacitor. Timer circuit using the integrated circuit 555 showed to be efficient component for a digital sensor based on timer circuit.

References

ARDUINO (2018), "What is Arduino? ";https://www.arduino.cc/en/Guide/Introduction.

Blick, B. (2002). "555 Timer as an A/D converter"; http://bobblick.com/techref/projects/a2d555/a2d555.html.

Cady, F. M. (1997) "Microcontrollers and Microcomputers principles of software and hardware engineering", Oxford University Press, New York.

Camenzind, H. (2017)**. “**The History of 555 Timer IC – Story of Invention by Hans Camenzind **“ ;** http://www.circuitstoday.com/the-history-555-timer-ic

RASPBERRY PI FOUNDATION (2018). “RASPBERRY PI DOCUMENTATION”, https://www.raspberrypi.org/documentation/.

Undergroundworkbench (2009); “555 Based Digital Voltmeter”; https://undergroundworkbench.wordpress.com/2011/02/21/555-based-digital-voltmeter/