Microcontroller based measurements : how to take out the best we can of them

VIOREL - CONSTANTIN PETRE
Faculty of Electrical engineering
Department of Instrumentation and data acquisition
University Politehnica
313 Spl. Independentei, sect. 6, Bucharest
ROMANIA

Abstract : - Few interesting aspects involved in microcontroller based measurement are approached along with solutions to take out all the benefits. Several ways of connecting the output signal of a transducer to a microcontroller are described and the case of a digital interpretable analog signal is examined. Conclusions are taken out right from application data and practical examples based on the author's experience in working with microcontrollers.

Key-Words : - Microcontroller, measurements, digital interpretable, capture, counting, cycles.

1 Introduction
It's obvious that there are a lot of microcontroller based architectures for measuring all sort of quantities. We can have a certain classification by taking a look at the microcontroller's role : either logical control unit, or measuring and counting unit, or communication unit, or anything else. Furthermore, one can observe that the microcontroller can have not only one but also many, or all of the mentioned functions; meantime, a measuring system can consist of two or more microcontrollers, each having particular tasks.
Among these wide possibilities, let's consider the situation where the microcontroller is directly taking care of the measuring process.

2 Microcontroller's tasks
As seen in fig. 2, the microcontroller can process a digital number (that is a very simple situation which rises no problems) or a digitally interpretable analogue signal. And that is the point where this paper will focus on.
The meanings of fig. 2 are :
x = measured (input) quantity,
y1 = digital number or a digital number transmitted over I2C or SPI protocols,
y2 = analogue quantity containing the information of the measured quantity,
y3 = digital number as output for A/DC,
A/DC = analogue to digital converter,
y4 = square wave output signal that can be interpreted by the user both analogue and digital,
μC = microcontroller.
First, we can see that is all about a time interval measurement but depending on transducer's output characteristic, we'll have to deal with period, pulse width or pulse and pause width.
Basically, the microcontroller has to accomplish two tasks :
- start / stop of counting,
- counting.
As simple as it looks but it is only now that the engineer / designer has to carefully analyse all the choices and pick-up the best solution.

2.1 Start / stop of counting
Let's try and see which are the possibilities :
a) sampling the signal and identifying the desired edge - fig. 1,

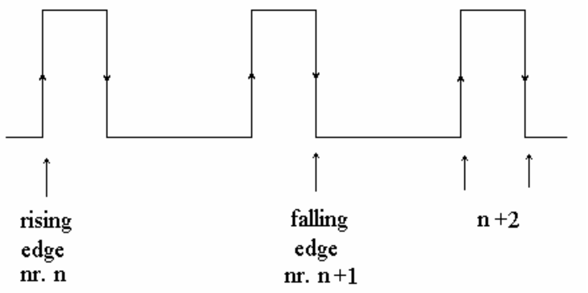

Fig. 1 Rising and falling edges of a signal

b) using the CCP facility (Capture-Compare- Pulse width modulation) of the microcontroller by capturing every rising or falling edge,

c) using the measured signal itself to start / stop an internal counter,
d) using the interrupt service of the microcontroller.
It derives that :
- there is no best solution but a best solution for a certain application,
- the CCP facility is not suited when both pulse and pause widths are to be measured,
- we can say that using the measured signal edges is appropriate only for large time intervals,
- as far as I am concerned, I would say that sampling the signal at the highest rate permitted by the microcontroller's clock is a very good choice that allows the user to measure the time interval with an accuracy of 2τ, where τ is the sampling interval or the instruction cycle.

For example, a microcontroller with 200 ns instruction cycle, will be able to measure time intervals with an absolute accuracy of 400 ns = 0.4 µs.

2.2 Counting

This process is related to the start / stop method used but basically there is a certain memory register being incremented (or decremented). We would say "basically" because for being very accurate we have also to count certain cycle instructions added and needed for each of the methods described for start / stop of counting.

Solution : At first look we would say that choosing a 16-bit, 32-bit, … microcontroller architecture will give us better performance. That is correct **but** there is also a **clue** here : why don't use a **cheaper** microcontroller to have the **same performance** ? Is it possible ?

Yes, **certain** manufactures provide the possibility to work with a RAM word variable (Random Access Memory), that is 2 bytes - so an 8-bit microcontroller will be able to measure time intervals just like a 16-bit one ! You can count not only at 255 but at 65,535. And this is a piece of secret known only by those working with that kind of microcontroller.

2.3 Start / stop and counting designed together

The last important aspect is related to the code written for the microcontroller or the software used to write it (like C, Basic, VisualBasic, VisualC, Pascal, Picbasic and many others).

Below is an example of a small Basic program for sampling the pulse width of a signal :

```
p = 0
1 a = INP(port)                #
  IF a = 127 THEN GOTO 1       # identify
2 b = INP(port)                # rising edge
  IF b = 111 THEN GOTO 2       #
3 p = p + 1                    ; begin counting
  c = INP(port)
  IF c = 127 THEN GOTO 3
 end
```

Many users prefer to work with a programming language and a compiler to microcontroller's code. It is more convenient but not in this case because there is strictly necessary **to know exactly the number of internal cycles** needed for executing the measuring process.

Here it is the same program as above, written with intrinsic instructions of a microcontroller :

```
        clrf 46
e1      btfss PORTA,0          #
        goto e2                # identify
        goto e1                # rising edge
e2      btfss PORTA,0          #
        goto e2                #
e3      incf 46,1              ; begin counting
        btfsc PORTA,0
        goto e3
        end
```

We can notice that the sampling cycle is well known, consisting of 4 instruction cycles.

Thus, we may use a programming language whenever the microcontroller has any other role in the system.

3 Conclusion

When a microcontroller is directly involved or is taking care of the measuring process, there are several approaches. Depending on the signal that has to be processed an optimum formula had to be used in order to ensure proper counting and also start / stop of it. As seen in the paper, for counting we need to have control over the number of internal cycles of the microcontroller and also we now know that a cheaper microcontroller can be able to offer more performance then it looks. As for start / stop of counting a certain feature of the microcontroller may be used, depending on the desired quantity to be measured.

References :
[1] Microchip Tecnology Inc. - 8/16-bit Microcontrollers, *Programming specifications*, 2006.

[2] Atmel Corporation - Automotive, AVR 8-bit RISC, 8051 Architecture, AVR 32, *Datasheets*, 2005.

[3] Intel Corporation - MCS51, MCS 96, *Technical documentation*, 2004.

[4] Philips Semiconductors - 8-bit 80C51, 16/32-bit ARM, *Reference manuals*, 2004.

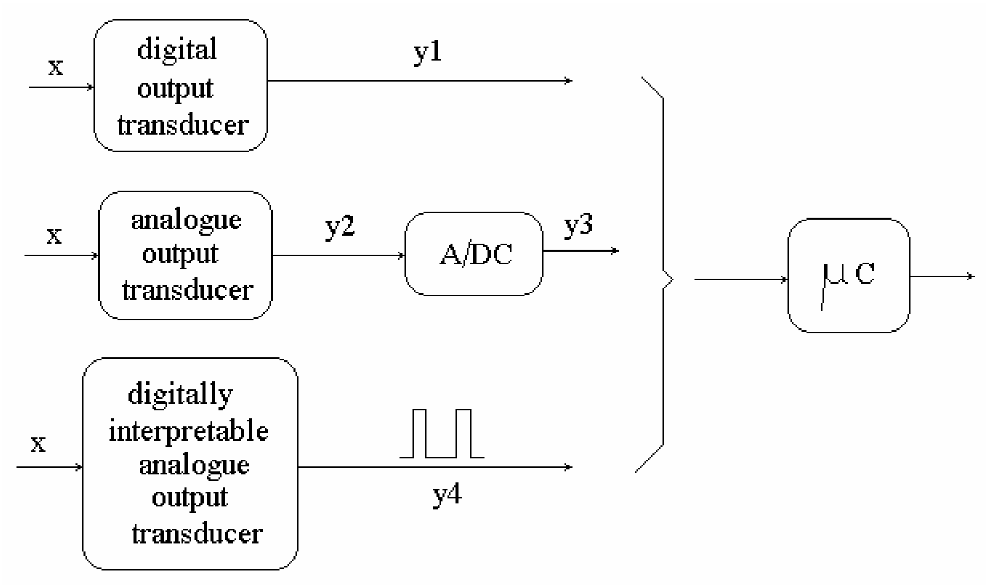
x
digital output transducer
y1
x
analogue output transducer
y2
A/DC
y3
μC
x
digitally interpretable analogue output transducer
y4

Fig. 1 Input quantities of a microcontroller within a measuring chain