Frequency Smart Sensor Data Acquisition

Joan Oliver

Departament de Microelectrònica i Sistemes Electrònics

Universitat Autònoma de Barcelona

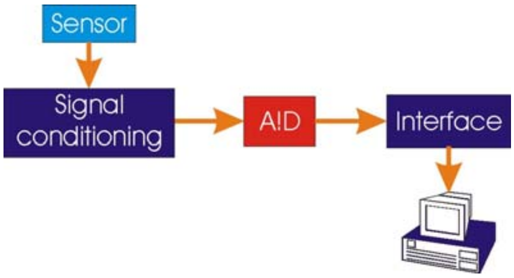
(joan.oliver@uab.es)

Outline

- Introduction
- Data acquisition in sensor systems
- Classical frequency to code methods
- Advanced frequency to code methods
- The software in the intelligent sensor systems
- Multichannel sensor systems

Introduction

- The classical microsystem signal processing is based in the conjunction of a set of devices that basically 'transports' the transducer signals to the computer.



Universitat Autònoma de Barcelona

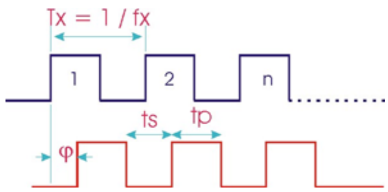
Smart sensor

- But MEMs technologies makes possible to integrate, in one substrate – monolithique-, or in several substrates –hybrid-, all the circuitry of the MEM: transductors, data processing and 'intelligence'.

Universitat Autònoma de Barcelona

Frequency-dependent sensors

- In the classification of sensors dependent of the output signal the frequency-behavioural sensors can be envisaged.
- The acquisition data circuitry of these sensors can be simplified and better quantisation errors obtained if specific frequency-to-time is considered. This is due to their *semi-digital* characteristics. Parameters to be considered, in this case, are:
  - fx = sensor frequency
  - Tx = 1/fx
  - tp = pulse width
  - ts = spacing interval
  - tp/(ts+tp) = duty-cycle
  - N = number of pulses
  - φ = phase shift



Frequency-dependent acquisition characteristics

- Frequency signal treatment has the following advantages:
  - High noise immunity to larger distances
  - High output power. Losses in frequency signals are minor in respect to those of continuous voltage dependent signals.
  - Dynamic range not dependent on noise or power supply voltages.
  - Accuracy is better. The measurement error can be programmed. Only systematic errors could be considered, but them can also be reduced using autocalibration.
  - Codification and interfaces easier.
  - Adaptability. The measurement parameters can be adapted to the measure conditions of the smart sensors
  - Trustworthy in the measure. It can be checked by autodiagnosis.

Sensor classification using frequency measurement: quasi-digital sensors

Stating the hypothesis Magnitude → Measure, we encounter the following

- Type A: x(t) → F(t)
  - It corresponds to sensors with direct frequency response
  - Circuitry only is needed in order to amplify and adapt impedances
  - Examples: resonants(piezzoelectrics, quartz resonators,…), coders, ...
- Type B: x(t) → V(t) → F(t)
  - They need a voltage/current – to- frequency converter.
  - Most sensors behave in this category: Hall sensors, photoelectric devices, photosensors, ...
- Type C: x(t) → P(t) → F(t)
  - Highly diverse sensor range
  - They should be signal modulator sensors based on the use of electronic oscillators that determine the output frequency response

Quasi-digital sensors

Quasi-digital sensors characteristics:

- Needless of AD converters
- Design process does not imply the usage of more stages that in respect to the traditional design using AD converters.
- Actual design tools already contains microcontrollers kernels, voltage-to-frequency converters (VFC), ... that makes faster the design of the measurement unit
- And, in general, the cost of the system is lower in respect to the circuitry used in AD converters systems

VFC versus ADC

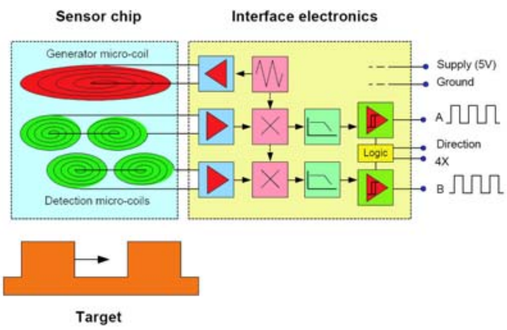
Advantages of the VFC versus the ADC are:

- Easier to build and cheaper
- Accurate with better linearity: useful in noisy environments
- On-line process of the received data
- Measurement errors could be greatly reduced
- Though conversion times are not very high, the use of pipeline processors allow the use of VFC in high frequency applications

Example: non-contact sensor for rotation speed

MS1200 micro-sensor (CSEM): Smart sensor for rotation speed based on inductive position, speed and direction, with

- CMOS compatible digital output , with frequency range: 0 – 40 KHz
- Core based on a chip with one generator coil and two detection coil sets
- The differential arrangement of the detection coils rejects common mode signal
- The outputs are two signals in quadrature plus a direction signal



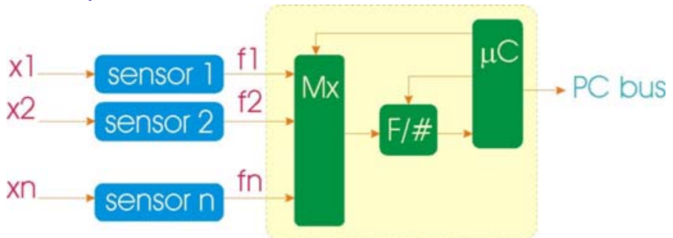
Data acquisition in sensor systems

- Actually, there are two main methods used in the information processing that arrives from the sensor array:
  - Temporal division methods, using data time multiplexing
  - Space-division methods using simultaneous acquisition data

Universitat Autònoma de Barcelona

Data acquisition using temporal division

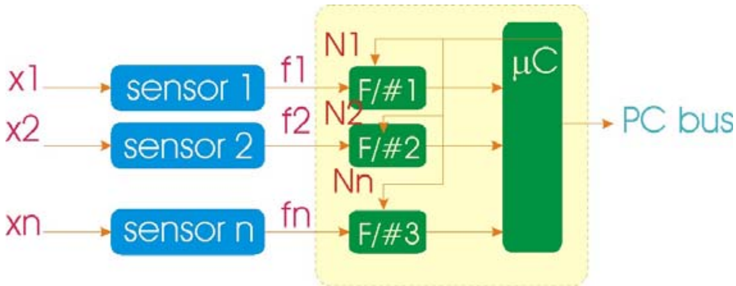
- **F/# → Frequency-to-code converter**
  - In a temporal window Tq (quantisation time) the F/# counts Tx=1/fx periods using a high precision f0 base (during one or more Tx periods)
  - Once data is acquired, data is processed in the uC or DSP, prior to be sent to the computer
  - The polling time over each sensor is given by
  - $\delta 0 = n \cdot (Tq + \delta delay1 + \delta delay2)$, where $\delta delay1$ and $\delta delay2$ are delays introduced by the connection time between sensors



Universitat Autònoma de Barcelona

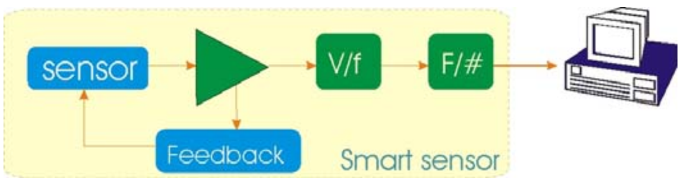
Space- division method

- Data sent to the µC by polling, interruption, DMA, ...
- N conversions can be made simultaneously
- Productivity and speed augmented by n times and determined by Tq + $t_{readout}$
- More acquisition channels implies additional hardware required



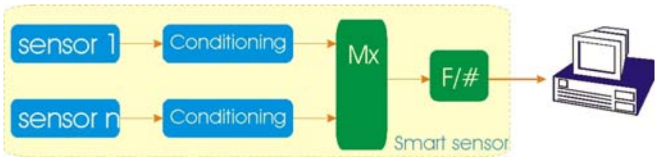
Smart sensors architectures (I)

- Acquisition data methods depends on the wired architecture
- Three main methods are used
  - Architectures with analogue process and frequency conversion

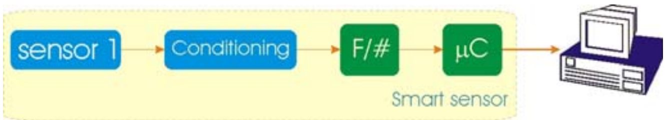


Smart sensors architectures (II)

- Smart sensor array



- µC based architectures
  - µC can store specific information about the sensor (non-linearities, ...) and make, also, the F/# conversion
  - Whole intelligent sensor can be integrated using standar libraries
  - PC communication can be established using serial, parallel, ... buses



Errors in multichannel data acquisition systems

- **The first error is in the same frequency sensor output due to the information quantisation**

  Conversion errors follow the normal distribution with $\delta_{max} \leq \pm 3\, \sigma_{sensor}$, where σ_{sensor} is the mean quadratic error, and δ_{max} the limit of the sensor error

- **The frequency measure introduces a new error, given by $\delta_{Tmax} = \delta_{trigger\text{-}err} + \delta_{0max} + \delta_{qmax}$. $\delta_{trigger\text{-}err}$, δ_{0max} and δ_{qmax} are errors of the clock trigger, the relative frequency and quantisation error, respectively.**

  That means

$$\sigma_T = \sqrt{\sigma^2{}_{trigger-err} + \sigma^2{}_0 + \sigma^2{}_q}$$

- **Another error is introduced in the computer due to the process (calculus) of the received data.**
- **So, the total error is given by**

$$\sigma_{DAQ} = \sqrt{\sigma^2{}_{DAQ} + \sigma^2{}_{f/\#} + \sigma^2{}_{Calculus}}$$

Classical F/# conversion methods

- There are more than 1000 patents related to F/# conversion methods.
- Between these methods, we will discuss the following:
- Classical methods
  - Standard direct counting method
  - Indirect counting method
    - Monocyclic methods
    - Bicyclic methods
  - Combined counting method
  - Phase shift-to-code conversion
- Advanced and autoadaptative methods
  - Proportional counting method
  - Reciprocal counting method
  - Dependent counting method
  - Relative values conversion method
  - Advanced phase shift-to-code method

Standard Direct Counting Method (I)

- Counting of Tx (fx unknown freq) cycles for a T0 (f0 reference freq) temporal window

$$Nx = \frac{T0}{Tx} = T0 \cdot fx \Rightarrow fx = \frac{Nx}{T0}$$

- Errors are mainly due to the absence of initial and final synchronisation.

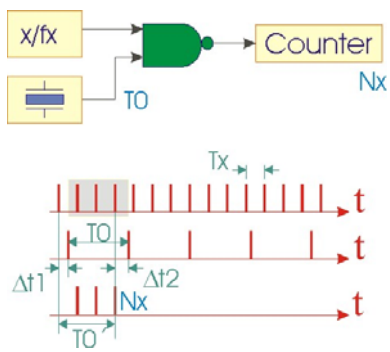
  These errors are Δt1 and Δt2. So, the actual measure time is given by

$$T0' = Nx \cdot Tx = T0 + \Delta t1 - \Delta t2$$

$$Nx = T0 \cdot fx \pm \Delta q,$$

where $\Delta q = (\Delta t1 - \Delta t2)/Tx$

$\rightarrow 0 \leq |\Delta q| \leq Tx$



Standard Direct Counting Method (II)

The measurement introduces two main errors:

- **Quantisation error:**
  - The maximum relative quantisation error is given by

$$\delta q = \pm 1/Nx = \pm 1/(T0 \cdot fx)$$

  - Then, the maximum absolute error is $-1 \leq \Delta q \leq 1$, and errors are distributed according to the triangular Simpson's distribution law

$W(\Delta q)=$	0 at $-1 > \Delta q > 1$	➔ mean error = $M(\Delta q) = 0$
	$1+\Delta q$ at $-1 \leq \Delta q \leq 0$	dispersion = $D(\Delta q) = 1/6$
	$1-\Delta q$ at $0 \leq \Delta q \leq 1)$	mean root square deviation = $s(\Delta q) = \pm D^{1/2} = \pm 6^{-1/2}$

  - Supposing that we can shift fx half a period in respect to T0 window, then $\delta q = - 1/(2 \cdot T0 \cdot fx)$

    That's equivalent to shifting half period the pulses of fx, what implies that $\delta q = \pm 1/(2 \cdot T0 \cdot fx)$ and, then

$W(\Delta q)=$	0 at $-0.5 > \Delta q > 0.5$	➔ mean error = $M(\Delta q) = 0$
	1 at $-0.5 \leq \Delta q \leq 0.5$	dispersion = $D(\Delta q) = 1/12$
		mean root square deviation = $s(\Delta q) = \pm D^{1/2} = \pm 12^{-1/2}$

Standard Direct Counting Method (III)

- The second error is introduced by the precision of the reference frequency, a systematic error: δref.

  Typical errors are in the order of $\delta ref = 10^{-6}$ to 10^{-8}
- Then, the absolute quantisation error is $\Delta q = \pm f0 = \pm 1/T0$ ➔

$$\Delta q max = \pm (\delta ref \cdot fx + 1/T0)$$

- And the relative quantisation error is

$$\delta q max = \pm (\delta ref + 1/(fx \cdot T0)) \cdot 100 \text{ (in \%)}$$

- It can be foreseen that the error is critical at low frequencies.
  - Example: Given $fx = 10Hz$ and $T0 = 1s$ => $\delta q=10\%$. A sampling time of $T0 = 1000s$ is needed in order to obtain a relative frequency error of $\delta q = 0.01$ %
- Reduction of δq can be achieved by changing the hardware.

Indirect Counting Method: period counting method (I)

- **It is an effective method at low frequencies**

$$Nx = n\cdot\frac{Tx}{T0} = \frac{f0}{fx} \Rightarrow Tx = Nx\cdot T0 \quad (\text{for } n = \#\text{cycles} = 1)$$

- **Quantisation error**

Tx = (Nx-1)T0+ Δt1 +(T0- Δt2)= NxT0+ Δt1 - Δt2= NxT0±Δq

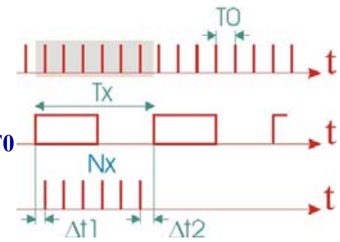
- **In this case, mean errors are W(Δt1)=0.5T0 and W(Δt2)=-0.5T0. They are equiprobable with probability 1/T0 →**
- **Maximum quantisation error = Δqmax = ± T0**

So

M (Δq) = M(t1) + M(t2) = 0,5 -0,5 = 0

D(Δt1) = D(Δt2) = D(Δt) = $T0^2/12$

σ (Δq) = $(\sigma^2(\Delta t1) + \sigma^2(\Delta t2))^{1/2} = \pm T0/6^{1/2}$

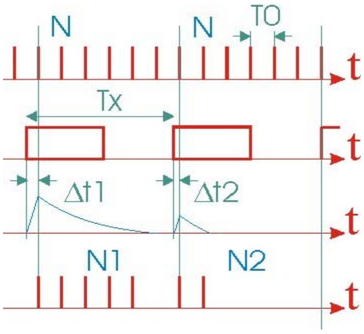


Indirect Counting Method: period counting method (II)

- **So, the conversion frequency is N_{fx} = 1/Nx**
- **There are two main methods used in order to obtain the conversion frequency:**
  - **Monocycle methods**
    - Usually fast and precise
    - Linearisation obtained with integrators with parallel carry or using decoders with ROM (that converts Nx to N_{fx})
  - **Bicycle methods**
    - Based on two steps:
      - at the first step they quantify from 1 to n periods.
      - in the second they compute NTx → N_{fx} using frequential multiplication / division
  - **Quantisation error : δq = ± (δref + 1/(f0·Tx·n)+δtriggererr/n)**
- **It is a bad method for high signal frequencies**

Indirect Counting Method: interpolation method (I)

- It reduces the quantisation error
- The method can exactly obtain the times Δt1 and Δt2 , responsible for the quantisation error
  - A capacitor begins to charge from the beginning of Tx until T0.
  - Follows the discharge of the capacitor at a ratio 1000 times slower
  - By counting the number of T0 pulses elapsed (N1, N2, ...), the quantisation error can be calculated and reduced.
  - Absolute quantization error = T0´=T0/1000
  - Speed determined by Tx and latency to new cycle



- However, at medium and high frequencies, the method has still important quantisation errors:

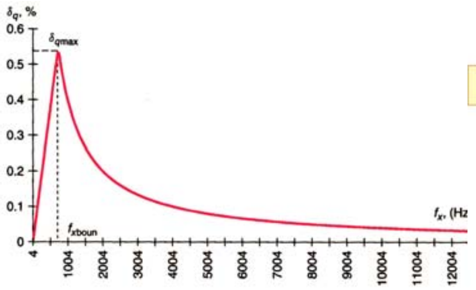
Example: for conversion of fx=10KHz and f0=1MHz, Δq=1%

Joan Oliver©. Bellaterra (Barcelona), October 2005

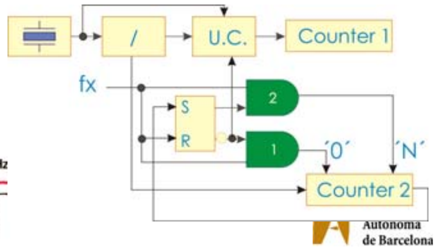
Universitat Autònoma de Barcelona

Combined counting method

- Adapted method: the standard counting method is applied at medium and high frequencies, and the indirect counting method is applied at low frequencies.
- The commutation limit (for n=1) is: fxbound = $(f0/T0)^{1/2}$
- Next plot shows the quantisation error, δq(fx) (f0=400/3 KHz, T0=0.25s)



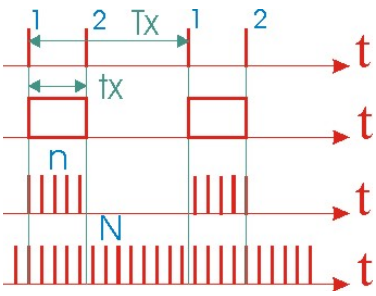
- Hardware can be built from figure. And2 is active at low frequencies, while at high frequencies And1 resets each cycle



Autònoma de Barcelona

Phase shift – to – code (φx-to-code) conversion

- Conversion φx-to-code computes the time interval on which two periodic pulse sequences are shifted
- The strobe is constituted by the pair of pulses 1 and 2, counting during their time interval, the number of f0 reference pulses.
  - In a tx interval there are n = f0 · tx pulses
  - In Tx there are N = f0 · Tx
  - The phase shift is given by φ = 360 · n/N
- Good resolution at low frequencies
- Conversion errors due to
  - Noise in the conversion process
  - Quantisation error
- Standard deviation is σ (Δq) = Δφq / $6^{1/2}$
- For a period, fxmax = Δφq / 360·T0
  - Examp: Δφq = 0.1º, f0=1MHz ➔ fxmax=277.7 Hz



Universitat Autònoma de Barcelona

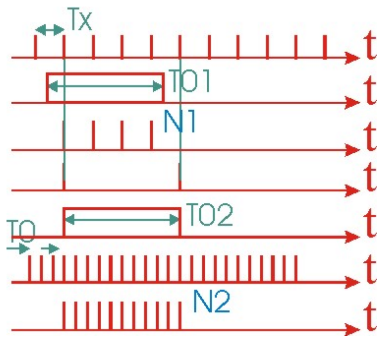
Advanced and autoadaptative freq-to-code methodes

- Advanced and autoadaptative methods
  - Ratiometric counting method
  - Reciprocal counting method
  - Dependent counting method
  - Relative values conversion method
- Conversion comparison
- Advanced phase shift-to-code method

Universitat Autònoma de Barcelona

Ratiometric counting method (I)

- Freq – to – code conversion with constant and small error on a wide frequency range
- It starts with an initial temporal window T01 and counts N1 pulses of Tx ➔ fx’ = N1/T01
- Once established T01, it synchronises with Tx pulses and forms T02, where T02 = N1·Tx. So, rounding errors are excluded.
- This second window fills with N2 pulses of f0 (second counter)
- Then N2 = N1 · Tx/T0 ➔ fx = f0 · N1/N2



Ratiometric counting method (II)

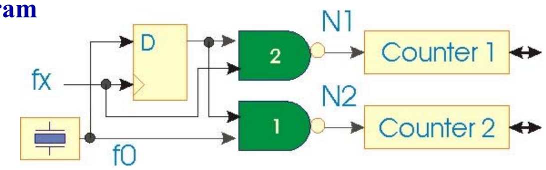
- Quantisation error independent of the conversion frequency and constant:

$$\delta 2 = \pm \frac{T0}{T02} = \pm \frac{T0}{N1 \cdot Tx} = \pm \frac{fx}{N1} \cdot T0 = \pm \frac{1}{f0 \cdot T01}$$

- Example: f0 = 1 MHz, T01 = 1 s ➔ $\delta q = \pm 10^{-4}$ %

  And, if interpolation method is applied measuring T02 = N1·Tx, then $\delta q = \pm 10^{-7}$ %

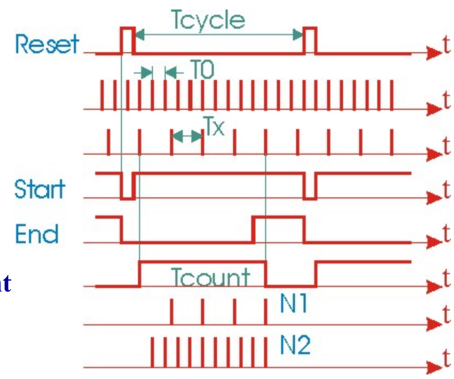
- Block diagram



Reciprocal counting method

- It is a variant of the last method
- Tcount begins with the pulse fx that follows the *start* pulse, and it stops with the fx pulse that follows the *end* signal
- Tx = T0 · N2/N1
- As with ratiometric counting method, this method has redundant time conversion
- The quantisation error is

$$\delta q = \pm 1 / (f0 \cdot Tcount)$$



Universitat Autònoma de Barcelona

Other advanced methods

- M/T counting method
  - Similar to previous methods, and differs in the synchronisation of the first reference time interval T01
  - Also overcomes standard methods demerits and achieves good resolution and accuracy
  - But still have redundant time conversion (like previous methods)
- Constant elapsed timer method
  - Another advanced method with constant quantisation error
  - Differs of the previous methods in the start and stop of the counters
- Single and double buffered methods, ...

Joan Oliver©. Bellaterra (Barcelona), October 2005

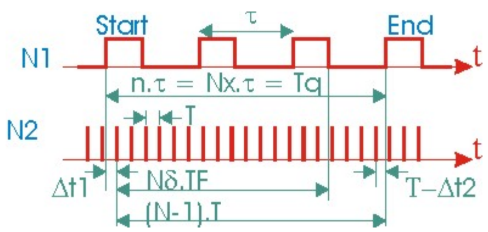
Universitat Autònoma de Barcelona

Dependent counting method (I)

- **Combines advantages of the classical methods and advanced methods with constant relative quantisation errors in a broad frequency range and high speed**
- **In can measure absolute and relative frequencies, intervals and deviations from specified values**
- **Next convention will be used**
  - **F as the greater of fx and f0**
  - **f as the minor of fx and f0**
- **The method is based on simultaneous**
  - **Separate counts of the periods of mesurand and reference frequencies according to the absolute and relative measurement method**
  - **Comparison of the accumulative number with Nδ (relative frequency measurement error specified by program)**
  - **Formation of a quantization window Tq equal to an integer number of periods Nx of the frequency f**
  - **Quantisation of the created reference time Tq during T F periods, so that $N \geq N_\delta$, being N the measurand ratio**

Dependent counting method (II)

- **Microcontroller synchronises the measure.**
- **It reads N at the moment a new τ impulse of the next period appears**
- **Counting conversions are replaced by computer operations**
  - **It executes during Tq**
  - **Measure can be done continuous, cyclic or one-time**



Dependent counting method (III): absolute values conversion

- **Given a specific**
  - Nδ, that is, given a δ error
  - a quantisation window Tq, that equals an integer number of τ pulses of f frequency
  - and a time that lasts for N cycles of the frequency F with a T period

pulses of frequencies fx and f0 are separately counted, and then

$$f = F\cdot\frac{n}{N} = F\cdot\frac{Nx}{N0} = F\cdot\frac{Nx}{N\delta + \Delta N}, \quad \tau = \frac{1}{f} = T\cdot\frac{N\delta + \Delta N}{Nx} \quad (1)$$

$$Tq = n\cdot\tau = Nx\cdot Tx = N\cdot T + \Delta t1 - \Delta t2 = N\cdot T \pm \Delta q \quad (2)$$

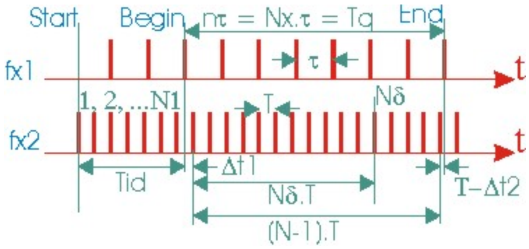
$$(measure\ conversion\ time)\ Tq = N\cdot T = (N\delta + \Delta N)\cdot T = \frac{1}{\delta}\cdot\left(1 + \frac{\Delta N}{N\delta}\right)\cdot T,\ with\ error < \pm T \quad (3)$$

In case τ = Tx (f0 ≥ fx) and T = T0 = 1/f0: $fx = f0\cdot\frac{Nx}{N\delta + \Delta N},\ Tx = T0\cdot\frac{N\delta + \Delta N}{Nx},\ Tq = \frac{T0}{\delta}\cdot\left(1 + \frac{\Delta N}{N\delta}\right)$

If τ = T0 (f0 < fx) and T = Tx: $fx = f0\cdot\frac{N\delta + \Delta N}{Nx},\ Tx = T0\cdot\frac{Nx}{N\delta + \Delta N},\ Tq = \frac{Tx}{\delta}\cdot\left(1 + \frac{\Delta N}{N\delta}\right)$

Dependent counting method (IV): conversion for relative values

- **It takes two steps.**
- **fx1/fx2 is performed without considering f0**
- **Step 1: Determination of the greater frequency**
  - **Calculus of N1=1/δ1, according to program-specified error in the bigger frequency**
  - **Separate counting**
  - **Summation of the impulse sequences periods Tx1=1/fx1 and Tx2=1/fx2**
  - **Comparison of both summations until to reach to N1 number**



Dependent counting method (V): absolute values conversion

- **Step 2: Measure of the fx1/fx2 (if fx1≤fx2) or fx2/fx1 (if fx1≥fx2)**
  - **Given the relative measure error δ, calculus of Nδ=1/δ.**
  - **Impulses summation of both sequences**
  - **Period-to-period comparison of the summation (pulse accumulation) of both frequencies with Nδ, according to the absolute dependent counting method**
- **Then** $\frac{f}{F} = \frac{n}{N} = \frac{Nx}{N\delta + \Delta N} \Rightarrow \begin{cases} \frac{fx1}{fx2} = \frac{Nx1}{Nx2 + \Delta N2}, \text{ for } fx1 < fx2 \\ \frac{fx2}{fx1} = \frac{Nx2}{Nx1 + \Delta N1}, \text{ for } fx1 \geq fx2 \end{cases}$
- **Quantisation time is not redundant and in limits:** $\frac{T}{\delta} \leq Tq = \frac{T}{\delta}\cdot\left(1 + \frac{\Delta N}{N\delta}\right) \leq \frac{2\cdot T}{\delta}$
- **Absolute error is:** $\Delta q = \frac{\Delta t2 - \Delta t1}{T}$
- **Relative quantisation error is:** $\delta q = \frac{\Delta q}{Tq} = \cdots \approx \cdots = \delta \cdot \frac{1}{1 + \frac{\Delta N}{N\delta}} \leq \delta q\max = \delta$

Dependent counting method versus standard method

Coefficient of variation of the quantisation error $\alpha = \frac{\delta\max}{\delta\min}$

- **DCM (fx=f, f0=F):** $\left.\begin{matrix} \delta\max = \frac{1}{N\min} = \frac{1}{N\delta} \\ \delta\min = \frac{1}{N\max} = \frac{1}{N\delta + \Delta N\max} \end{matrix}\right\} \Rightarrow \alpha = \frac{N\delta + \Delta N\max}{N\delta} = \cdots = 1 + \frac{1}{N\delta}\frac{F}{f} = 1 + \frac{f0}{fx}$

Example 1: Let's the case fx = 2Hz, f0 = 10MHz, Nδ = 10⁶ (δ = 10⁻⁶·100% = 0.0001%)

$$\alpha = 1 + \frac{1}{10^6}\frac{10^6}{2} = 1.5, \text{ where } \begin{cases} \delta\max = 10^{-6} \\ \delta\min = \frac{2}{3}\cdot 10^{-6} \end{cases} \Rightarrow \begin{cases} \text{qerr max is constant} \\ \alpha < 1.5\cdot\delta_{\max} \text{ for } f = 2\cdots 10^6\,\text{Hz} \end{cases}$$

For the same measuring period, SDCM or ICM, α will be about 500000

Example 2: If fx = 2·10⁴Hz, f0 = 1MHz, Nδ = 10⁶ (δ = 10⁻⁴%)

According to dependent count method: $tx = \frac{10^6 + 10^6/2\cdot 10^4}{10^6} \cong 1\,\text{sec}$

Using standard direct counting method: $tx = \frac{1}{\delta\cdot fx} = \frac{N\delta}{fx} = \frac{10^6}{2\cdot 10^4} = 50\,\text{sec}$

Using indirect counting method: $NT = \frac{fx}{\delta\cdot f0} = \frac{2\cdot 10^4\cdot 10^6}{10^6} = 2\cdot 10^4 \Rightarrow tx = NT\cdot Tx = \frac{2\cdot 10^4}{10^4} = 2\,\text{sec}$

Conversion comparison ranges

- **Conversion ranges of the methods are:**

f (quantization error, conversion time) F

Dependent count (1/Nδ, Nδ + ΔN/f0)

Ratiometric (1/(f0·T01), T01+ ΔT)

Reciprocal (1/(f0·Tcount), Tcount+ ΔT)

M/T counting (1/(f0·T01), T01+ ΔT)

Constant elapsed time (1/(f0·T01), T01+ ΔT)

Single/Double buffered (1/f0·T0, T0)

DMA transfer (1/(f0·T0), T01+ ΔT)

Indirect counting (fx/f0,Tx)

Quantisation error Conversion time

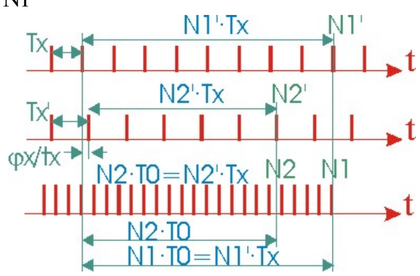
Direct counting (1/(T0·fx),T0)

Advanced phase shift – to – code (φx-to-code) conversion

- **Method of the coincidence : phase shifting between two periodic pulse sequences of Tx period: N1 pulses of T0 and N1' pulses of Tx That is:**

$$\left.\begin{array}{r} N1 \cdot T0 = N1' \cdot Tx \\ N2 \cdot T0 = N2' \cdot Tx + tx \end{array}\right\} \Rightarrow \left\{\begin{array}{l} \varphi x = \dfrac{N1' \cdot N2 - N2' \cdot N1}{N1} \cdot 360 \\ tx = \dfrac{N1' \cdot N2 - N2' \cdot N1}{N1} \cdot t0 \text{ (conversion time)} \end{array}\right.$$

- **Measure absolute error only depends on tx. It can be reduced up to $0.1 \cdot 10^{-12}$**
- **Phase shift conversion absolute error could be of 0.05º at 1MHz**



Software in the intelligent sensor systems

- Introduction
- PCM in the proportional counting method
- PCM error analysis
- Systematic error correction
- Use of µC in smart systems
- ...low power through software optimisation

Introduction

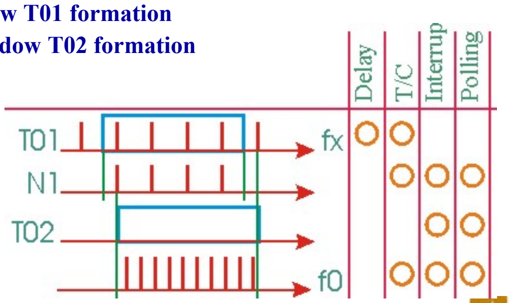
- Frequency output sensors are prone to be digitally processed, usually by use of Program Conversion Methods (PCM)
- Then errors are inherent to the sensor
- The goal is to obtain precise, low power and autoadaptative PCMs
- So, a PCM is a

Processor algorithm of measurement incorporated in the functional logic structure of the microcontroller through the software

- PCMs could be complex programs as they might incorporate temporal critic routines
- Classical routines implemented in microcontrollers that are responsible for the frequency to code conversion should work using
  - Polling for the counting
  - Counting with timers/counters
  - Using microcontroller interruptions

PCM in the proportional counting method (I)

- Traditionally, the calculus required for the frequency to code conversion has been a drawback of these methods
- The calculus power of actual microcontrollers has overcome this problem
- The solution is to divide the measure in elemental components that can be executed concurrently:
  - First temporal window T01 formation
  - Second temporal window T02 formation
  - f0 counting
  - fx counting
- Solutions could be



PCM in the proportional counting method (II)

Delay

- It can be built from nested loops with well tabulated delays
- Minimum elapsed time fixed by the shortest µC instruction time (usually that of the NOOP instruction)
- As a drawback, it wastes µC resources

Timers/counters

- Limited by the maximum counting of the counters
- The start and stop systematic errors must be computed

Polling

- It is simple to perform. For example, asking for a '1' in an input of the µC
- In order to avoid fx pulses loses, the maximum conversion frequency (1/Txmax) and the pulse width (τx) must accomplish
  - $Txmax \geq n \cdot \tau cycle$
  - $\tau jump \leq \tau x \leq n \cdot \tau cycle$

  where τcycle and τjump stands for the instruction time and jumping execution time, respectively

Interruptions

- Can be applied when $Txmax \geq Tint \geq \tau cycle \cdot n$, Tint = routine interruption time spent

PCM in the proportional counting method (III)

- **The conversion algorithm mus be metrologically efficient. Conversion errors must attend to**
  - **Methodic errors, due to the processing algorithms**
  - **Measurement circuitry computing capacity**
  - **Programming styles**
- **The election of a PCM could not be an easy task**

  For example, the proportional counting PCM algorithm has to perform three tasks in parallel:
  - Built the T01 temporal window for frequency pulse counting
  - fo frequency pulses counting
  - fx frequency pulses counting

  And, taking into account incompatibilities of the plausible architectures, considering
  - 1 timer/counter + 1 interruption ➔ $Vn = \{V_3^n\} = C_3^1 \cdot C_3^1 \cdot C_2^1 - 8 = 10$ solutions
  - 2 timer/counter + 1 interruption ➔ 17 possibilities
  - 3 timer/counter + 1 interruption ➔ 18 solutions

  So, it is really important the election of the µC

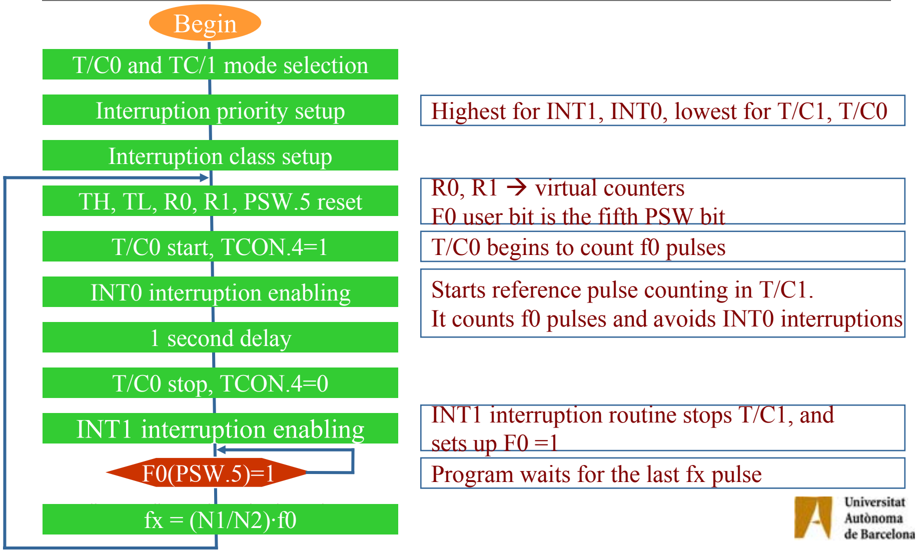
PCM in the proportional counting method (IV)

- **Given the µC and the different possible PCMs, the final PCM election is a function of the quantisation error (δ (%)) and the maximum frequency conversion fxmax**

  Is smart sensors, the ROM memory size S_{ROM} and power consumption P are also important parameters
- **However, and due to the great number and different architectures of the actual µC, the election of the final PCM reduces to the realisation of an optimum algorithm**
- **Example**

  PCM diagram flux based on an MCS-51 Intel family microcontroller that can afford to the algoritm
  - 2 timers (T/C0 and T/C1) → that counts fx and f0
  - software delay → T01 window formation
  - Interruptions (using counter overflow) → using registers R0 and R as a virtual counters → allow maximum frequency conversion with minimum quantisation error
  - T02 formed using two level interruption on INT0 and INT1

PCM in the proportional counting method (V)



Universitat Autònoma de Barcelona

PCM error analysis (I)

- **Errors in frequency-to-code conversion**
  - **Reference errors**
    - Frequency reference error
    - Temporal reference error
  - **T02 window formation**
    - Front wave formation error
    - Tail wave formation error
    - Interruption request delay error
  - **Computing errors**
    - Numbering representation error
    - Rounding/truncating error

Universitat Autònoma de Barcelona

PCM error analysis (II): reference errors

- The conversion accuracy is a function of the quartz crystal stability. Non compensated quartz crystal frequency deviation are 1÷50 · 10^{-6} T. Standard total incertitude in the µC measure is about $11 \cdot 10^{-6}$.
- Temperature compensated oscillators can improve this measure

PCM error analysis (III): computing errors

- In the fx(Tx) computing there are multiply and division operations. For example, in the proportional counting fx = f0·N1/N2. Due to the reduced number representation (8 to 16 bit, usually), errors in the enumeration and rounding errors must be considered.
  - Being B the base of the numbering system, the representation error of a number X is

$$\sigma_{errorX} = \frac{1}{2} B^{-n0}$$

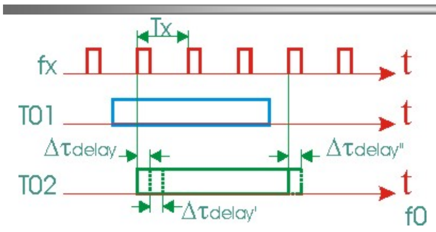
- fx computing must be made in fixed point representation. Some rules to minimize errors are:
  - Perform operations with numbers without sign
  - Substitute, whenever it is possible, multiplication by shift
  - Use residue restoring division
  - If f0=constant ∉ Z, use a scale factor km / K = f0·km = constant and

$$fx = \frac{N1 \cdot (f0 \cdot km)}{N2 \cdot km} = \frac{N1 \cdot K}{N2 \cdot km}$$

  - To minimize errors, the order of the operations must be
    N1·K = k1
    N2·km = k2
    fx = k1/k2,

  and fx ∈ [fxmin, fxmax] must be accomplished

PCM error analysis (IV): T02 formation error



$\Delta\tau$delay = tCALL + tJUMP + tSTART

$\Delta\tau$delay = systematic error

tSTART = start timer/counter

$\Delta\tau$delay' $\in$]0; τcmax], τcmax = 2· τcycle → Answer time due to the instruction arrival. Usually it is executed at the end of the actual instruction

$\Delta\tau$delay" $\in$ [3· τcycle ; 5· τcycle] → It depends on the last polling instruction execution

- ➔ So,
- ➔ T02real = T02 - $\Delta\tau$delay - $\Delta\tau$delay' - $\Delta\tau$delay"

PCM error analysis (V): PCM systematic error corrections

- Methodical errors and instrumentation errors can be reduced using actual (fast) µC, improving frequency-to-code algorithms and implementing fast and accurate data processing.
- PCM errors can be reduced taking in consideration good protocols in the PCM design step
  - For example, the interruption call can be changed:
    - Usually, the unconditional jumping is included in the interruption vector
    - But it can be first included the timer/counter starting and, next, the unconditional jumping, and then
      - $\Delta\tau$delay = tCALL + tSTART = 3· τcycle
    - Next, $\Delta\tau$delay can be further improved correcting T0 period by software (because, now, the systematic error is known)
- In general, systematic errors could be reduced by software

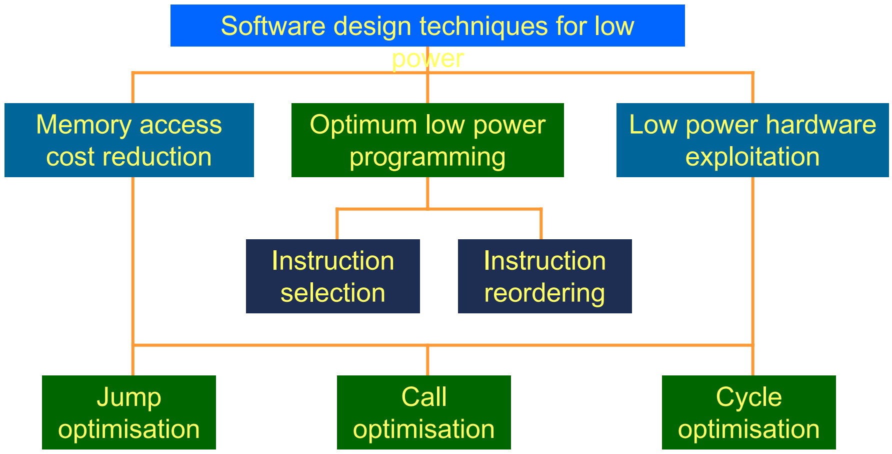
Use of µC in smart systems

- µC continue being used in measuring instrumentation due to their versatility, low cost and as a cheap debugging tools.
- However, standard µC usually are instruction redundant for instrumentation measure purposes
- A study made over the Intel 8051 instruction set used in measure instrumentation tasks reveals that
  - Only 27 instruction groups are used over the total instruction set
  - 9 from those are used in the 81% of the cases
  - The more used instructions are: MOV (51%), LCALL, MOVX, NOP, AJMP, DJNZ, ANL and RET
- If the instruction set not used were keep off the µC, the savings in die area would be 1/3, and savings on power consumption also would be significant

Low power design techniques in embedded microcontrollers cores (I)

- The µC could be the responsible, in a great measure, of the conditioning signal circuitry. Sensors, µC, frequency-time converters, signal conditioning circuitry can be joined on a substrate forming the smart sensor and thus lowering power consumption.
- In fact, low power design circuits are being critic processes, especially in telemetry systems. In this line, it is not solved the power reduction at instruction level (software).
- In that sense, power reduction can be effective
  - In compiler optimisations
  - In good algorithm design for power minimisation

Low power design techniques in embedded microcontrollers cores (II)



Low power design techniques in embedded microcontrollers cores

- **The μC could be the responsible, in a great measure, of the conditioning signal circuitry. Sensors, μC, frequency-time converters, signal conditioning circuitry can be joined on a substrate forming the smart sensor and thus lowering power consumption.**
- **In fact, low power design circuits are being critic processes, especially in telemetry systems. In this line, it is not solved the power reduction at instruction level (software).**
- **In that sense, power reduction can be effective**
  - **In compiler optimisations**
  - **In good algorithm design for power minimisation**

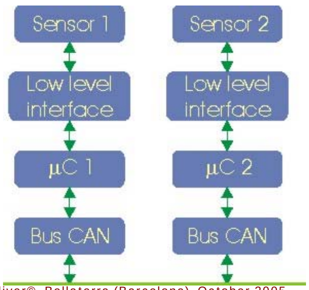
Multichannel sensor systems (I)

Hardware and software are the bridges between analogue systems and digital signal processing.

- **Monochannel sensor interface**

  A multilayer system should be used in smart systems (like in cars).

  Resolution and accuracy are obtained in the frequency measurement



- **Multichannel sensor interface**

  Parallel process information (from several sensors) implies the sequentiation of elemental measure procedures.

  Usually constraints on multisequentiation are imposed by the temporal timings of operations.

  The dependent counting method should be used in these cases, as it ensures non-redundant frequency-to-code conversion.

Car ABS system: A case study (I)

- **Wheel block causes a missfunctioning of the brakes, specially in the case of sliding surfaces and thus removing car security**
- **In the car, four sensors measure the car velocity, diminishing the brakes pressure as it approaches to 0.**
- **ABS reliability depends on the data processing speed (at real time). Speed rotation measure should be made using the direct or indirect counting or with the proportional method. So, it includes:**
  - **Rotational speed sensor**

    For ABS should be used: Hall sensors, modulators (active semiconductors) sensors, and passive inductance autogenerative sensors
  - **Coding**

    Using sensors as that of Hall effect, the output directly generates the pulses. So, in this case is simple
  - **Autoadaptative method for rotational measures**

    The dependent counting method should be used, as it gives the minimum measure time with quantisation time independent of the rotational speed.

    The sampling of the sensors output ensures that the information arrives at the four channels at the same time

Car ABS system: A case study (II)

- **Sampling time:**
  - **Supposing**
    - The car velocity of 240 Km/h
    - tmeasurement < 0.1 seg → The car will drive no more than 8.7 m/s
    - Relative error of measurement between 0.05% and 0.5%, with f0 = 1/T0 = 1 MHz
  - **Then the time of quantisation is given by**

    T = (1÷2)·T0·N = (1÷2)·T0/δ, N = minimum number of impulses

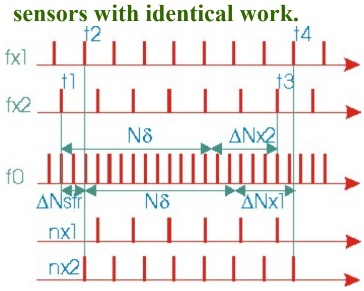
    → with δ = 0.05% → T = 2 ÷ 4ms → 25 ÷ 50 samples/sec

    → with δ = 0.5% → T = 0.2 ÷ 0.4ms → 250 ÷ 500 samples/sec

  - **If f0 = 10 MHz, the time can be reduced by a factor of 10**
  - **Using high rate sampling, the measure time can be reduced by 10 without altering the accuracy**
  - **The measure system can also measure the rotational acceleration**

Car ABS system: A case study (III)

- **In order to rationally distribute the microcontroller resources and to simplify the multichannel measure, the four channels are divided in two pairs of sensors with identical work.**



fx1 > fx2, and both frequencies acquired by interruption

f0 = Fosc/12, Fosc = µC clock

Nδ = 1/δ is stored in a timer/counter

(channel 1) Timer/counter begins to count in t1, with frequency f0. In parallel also fx2 is counted

Channel 2 works similar

- **Then frequencies of each channel are counted (with δ1 and δ2 specified ranges)**

$$\left.\begin{aligned} f_{x1} &= \frac{n_{x1}}{N_\delta + \Delta N_{x1}} \cdot f_0 \\ f_{x2} &= \frac{n_{x2}}{N_\delta + \Delta N_{x2}} \cdot f_0 \end{aligned}\right\} \rightarrow \left\{\begin{aligned} t_{x1} &= \frac{n_{x1}}{f_{x1}} = \frac{N_\delta + \Delta N_{x1}}{f_0} = \frac{1}{\delta_1 \cdot f_0} \\ t_{x2} &= \frac{n_{x2}}{f_{x2}} = \frac{N_\delta + \Delta N_{x2}}{f_0} = \frac{1}{\delta_2 \cdot f_0} \end{aligned}\right.$$

References

- N.V. Kirianaki, S.Y. Yurish, N.O. Shpak, V.P. Deynega. *Data Acquisition and Signal Processing for Smart Sensors*. John Wiley & Sons. 2002

Thank you for your attention

Joan.Oliver@uab.es

Joan Oliver©. Bellaterra (Barcelona), October 2005

Universitat Autònoma de Barcelona

Appendix: Simpson's (triangular) distribution

- **Empleado básicamente en Economía y es útil en problemas en los cuales se conocen muy pocos o ningún dato.**
- **Esta distribución tiene 3 parámetros, a (límite inferior de la variable); b (el modo) y c (límite superior de la variable).**
- **La esperanza es (a+b+c)/3 y la varianza es (a(a-b)+c(c-a)+b(b-c))/18**
- **Aplicaciones : La distribución triangular se define luego que se conocen los 3 parámetros a, b y c .**
- **Nos permite estimar las duraciones de las actividades de un proyecto usando las tres estimaciones : optimista, muy pesimista, y pesimista.**
- **Su función de densidad tiene forma triangular, y viene definida por f(x)**
- **Se denomina triangular cuando viene definida por dos parámetros, el valor mínimo y el valor máximo. En este caso el triángulo es equilátero. Se denomina triangular (triangular general), cuando viene dada por tres parámetros, el valor mínimo y el valor máximo de la variable, y el valor del punto en el que el triángulo toma su altura máxima. En este caso el triángulo no es necesariamente equilátero.**
- **Densidad representa la función de densidad de la distribución triangular viene dada por**

$$f(x) = \begin{cases} \dfrac{2}{a(a+b)}(a+x) & -a < x < 0 \\ \dfrac{2}{b(a+b)}(b-x) & 0 < x < b \end{cases}$$

$$\text{Densidad} : \begin{cases} \dfrac{2(x-a)}{(c-a)(b-a)} \text{ si } a \le x \le b \\ \dfrac{2(c-x)}{(c-a)(c-b)} \text{ si } b \le x \le c \end{cases}$$

$$\text{Media} : \frac{a+b+c}{3}$$

$$\text{Varianza} : \frac{a^2+b^2+c^2+ac-ab-bc}{18}$$