

# **Física Experimental VI – 4300314**

**2º Semestre de 2017**

**Instituto de Física  
Universidade de São Paulo**

**Professor: Antonio Domingues dos Santos**

**E-mail: [adsantos@if.usp.br](mailto:adsantos@if.usp.br)**

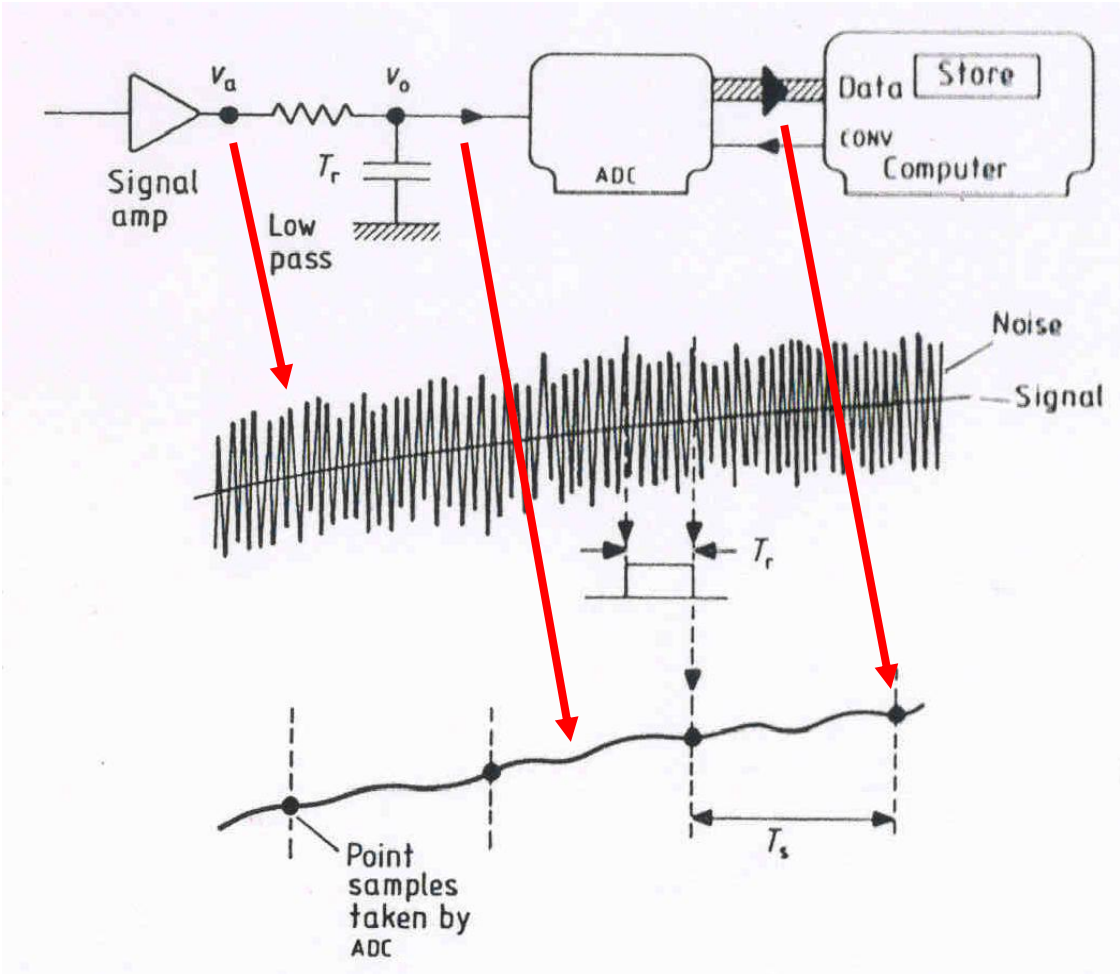
**Fone: 3091.6886**

# Ruído e Digitalização

## Sistema típico de digitalização de sinais

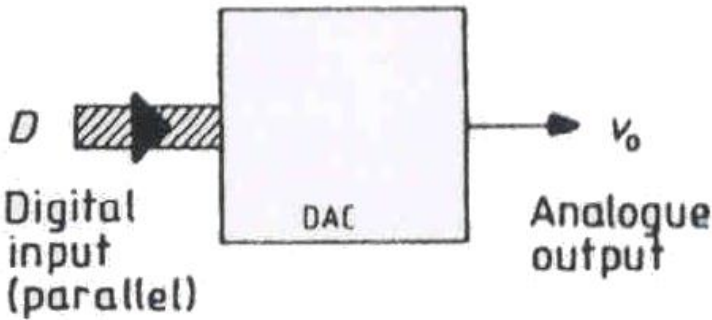
Para otimizar o desempenho e evitar desperdício de tempo,  $T_r$  deve ser igual a  $T_s$

Acréscimo no ruído dado por  $(T_r/T_s)^{-1/2}$



# Ruído e Digitalização

## Conversor Digital-Analógico (DAC)



### Amplificador Somador

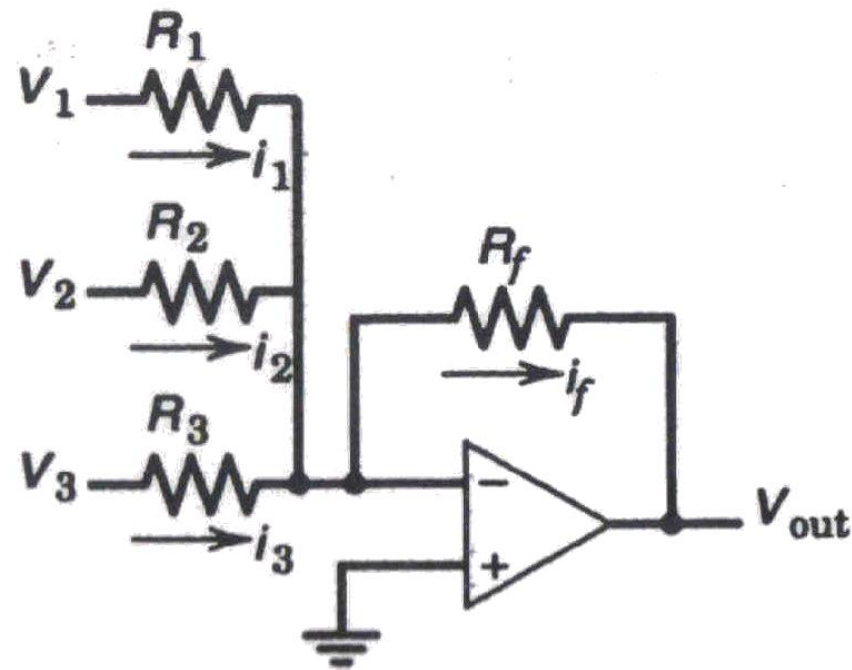
Onde o número de entradas corresponde ao número de bits

$$V_{out} = -\frac{R_f}{R_0} \cdot V_0 - \frac{R_f}{R_1} \cdot V_1 \dots - \frac{R_f}{R_7} \cdot V_7$$

Considerando-se  $V_0=V_1=\dots=V_7= 0$  ou 1

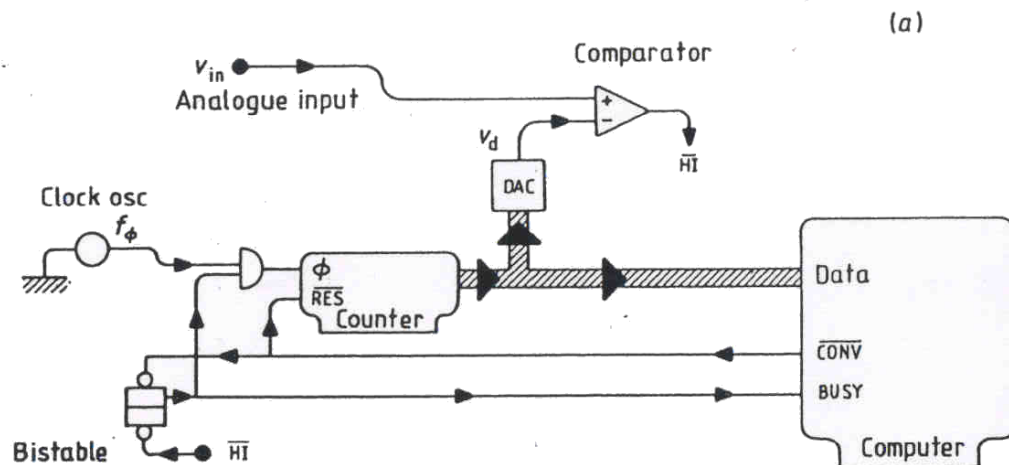
$$V_{out} = -R_f \sum_{i=0}^7 \frac{V_i}{R_i}$$

Onde:  $R_i=2^iR$



# Ruído e Digitalização

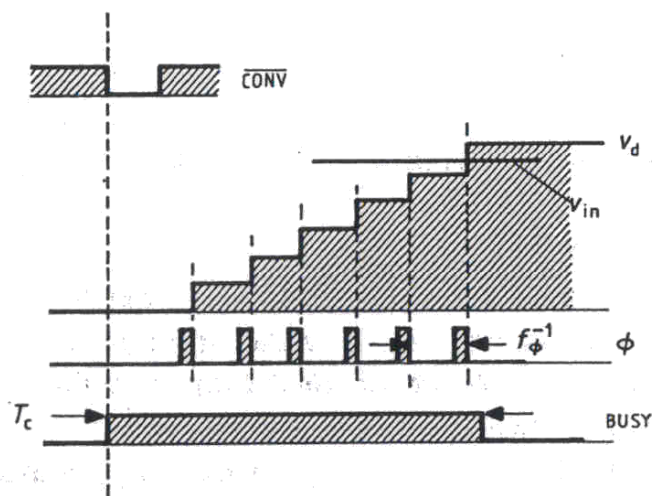
## Conversor Analógico-Digital (ADC)



### 1) Tipo “Contador”

O contador soma +1 a cada pulso do “clock”.

Quando  $V_d > V_{in}$ , para-se a contagem e armazena-se o dado.

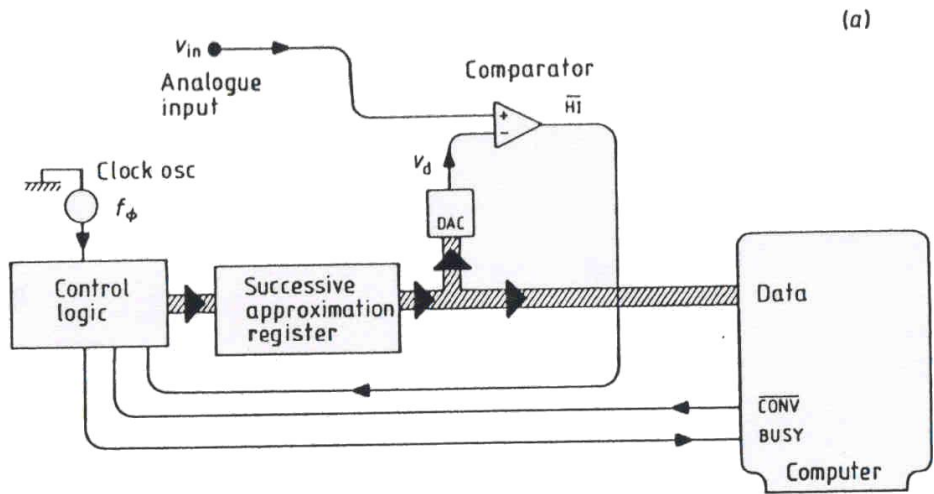


Para 8 bits e  $f = 1 \text{ MHz}$ , o tempo máximo de amostragem é

$$T_c = 1 \mu\text{s} \cdot 2^8 = 256 \mu\text{s}$$

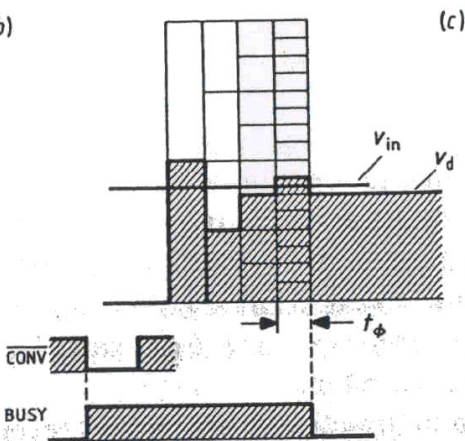
# Ruído e Digitalização

## Conversor Analógico-Digital (ADC)



(b)

```
Clear all bits of register
For i = 3 to 0
  set bit i
  if HI = 1 then reset bit i
```



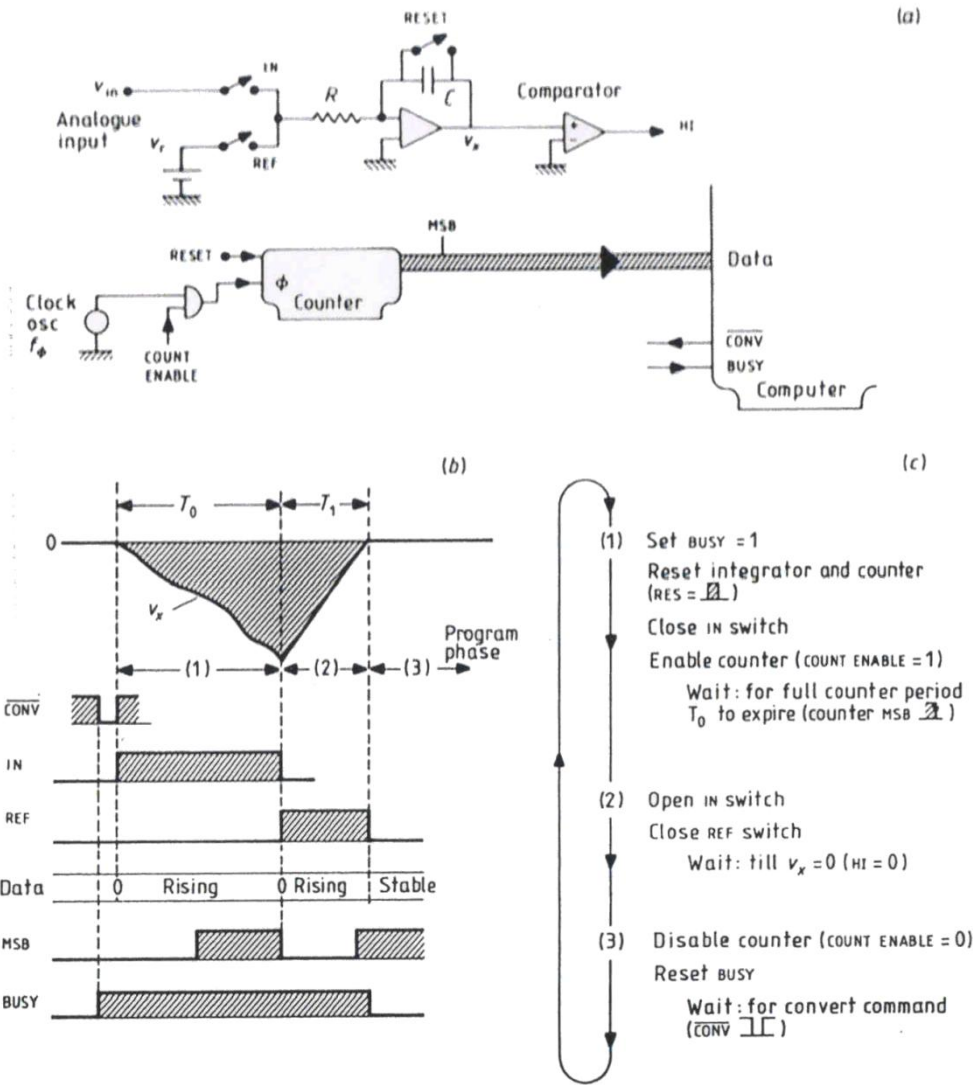
2) de “Aproximação sucessiva”  
Coloca-se sequencialmente cada bit em 1 e verifica-se se  $V_d > V_{in} \rightarrow \text{bit} = 0$ .  
Após todos os bits serem testados, armazena-se o dado resultante.

Para 8 bits e  $f = 1 \text{ MHz}$ , o tempo máximo de amostragem é  
 $T_c = 1 \mu\text{s} \cdot 8 = 8 \mu\text{s}$

Mais rápido !!!

# Ruído e Digitalização

## Conversor Analógico-Digital (ADC)



3) de “Rampa dupla”

Integra-se  $V_{in}$  durante o tempo  $T_0$  e integra-se  $V_r$  durante o tempo  $T_1$ , quando o integrador volta a zero.

$$V_{in} = T_1 V_r / T_0$$

$$V_x = -\frac{1}{RC} \int_0^T V_{in} dt$$

Para 8 bits e  $f = 1\text{ MHz}$ , o tempo máximo de amostragem é

$$T_c = 1\mu s \cdot 2^8 = 256\mu s \text{ (x2)}$$

Mais lento !

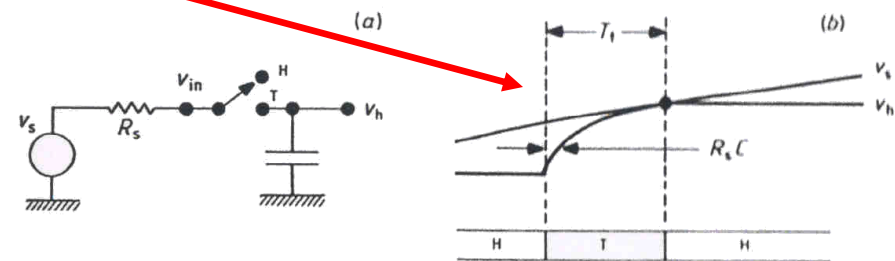
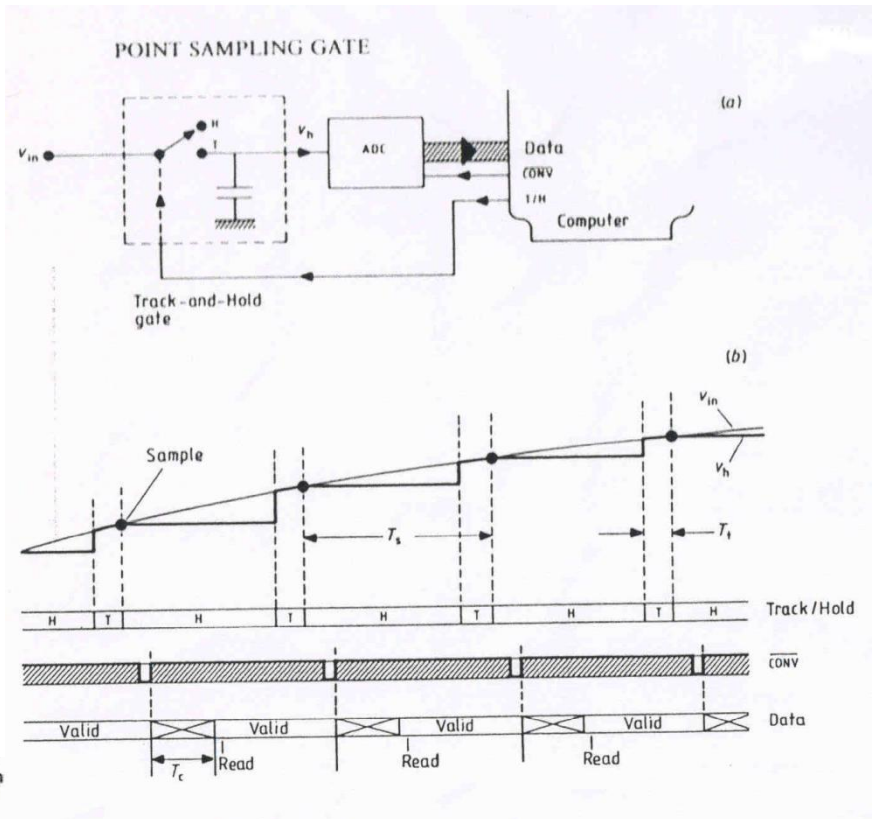
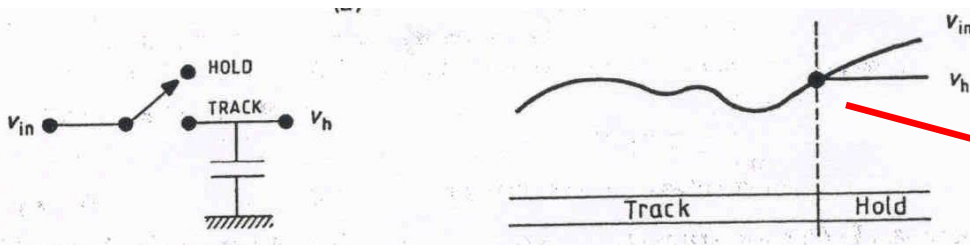
Mais preciso !

# Ruído e Digitalização

## Conversor Analógico-Digital (ADC)

Para ADCs “Contadores” e de “Aproximação sucessiva”, se  $V_{in}$  não é constante, haverá flutuações de medida (ruído extra!).

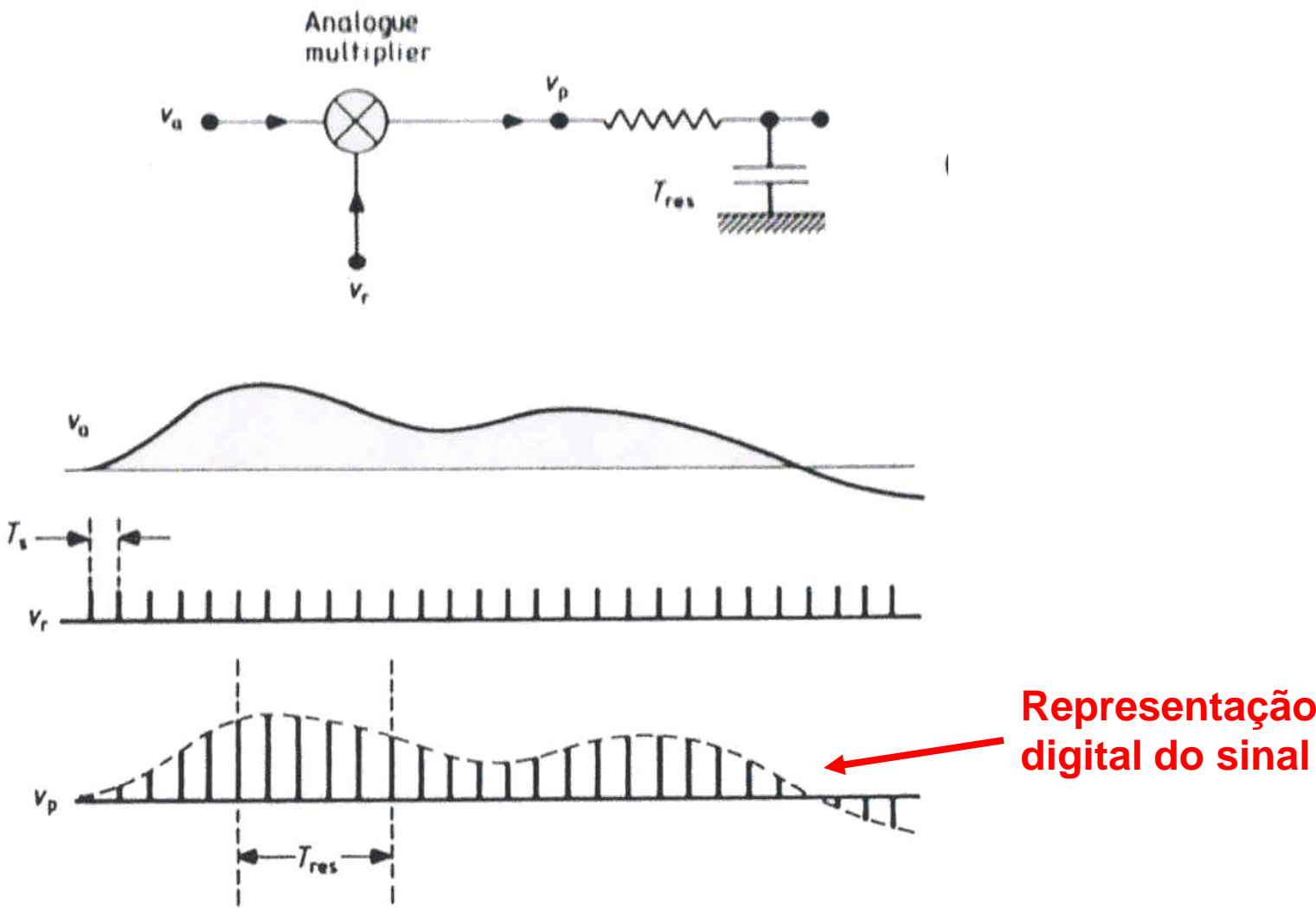
Tornando-se conveniente o uso de “amostragem e sustentação” (track-and-hold)





# Ruído e Digitalização

## Conversor Analógico-Digital (ADC)

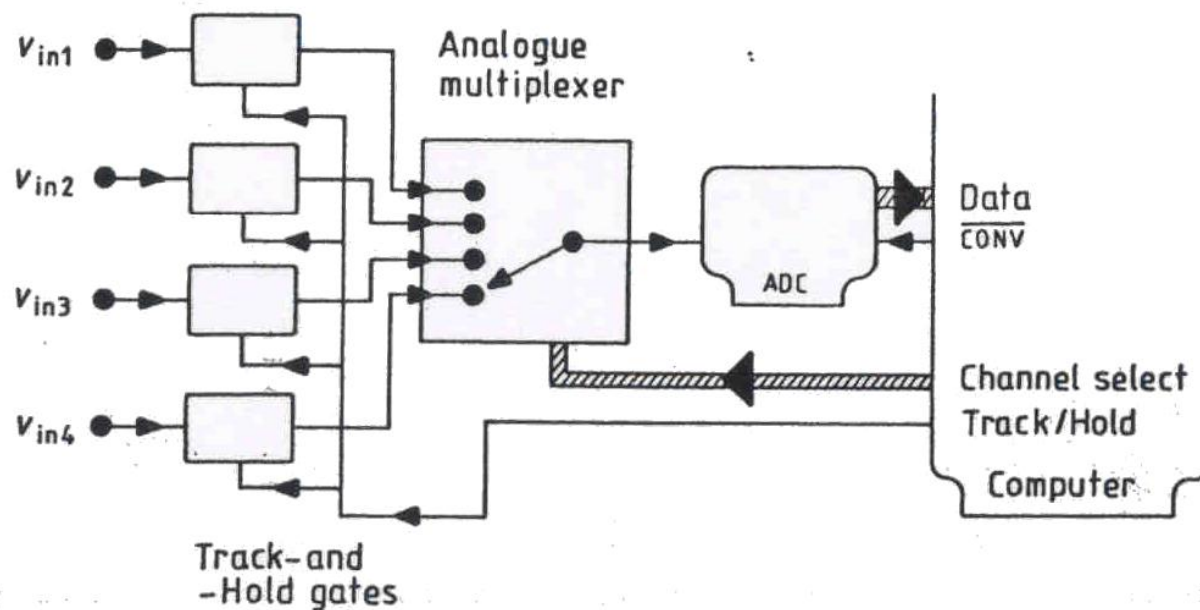




# Ruído e Digitalização

## Conversor Analógico-Digital (ADC)

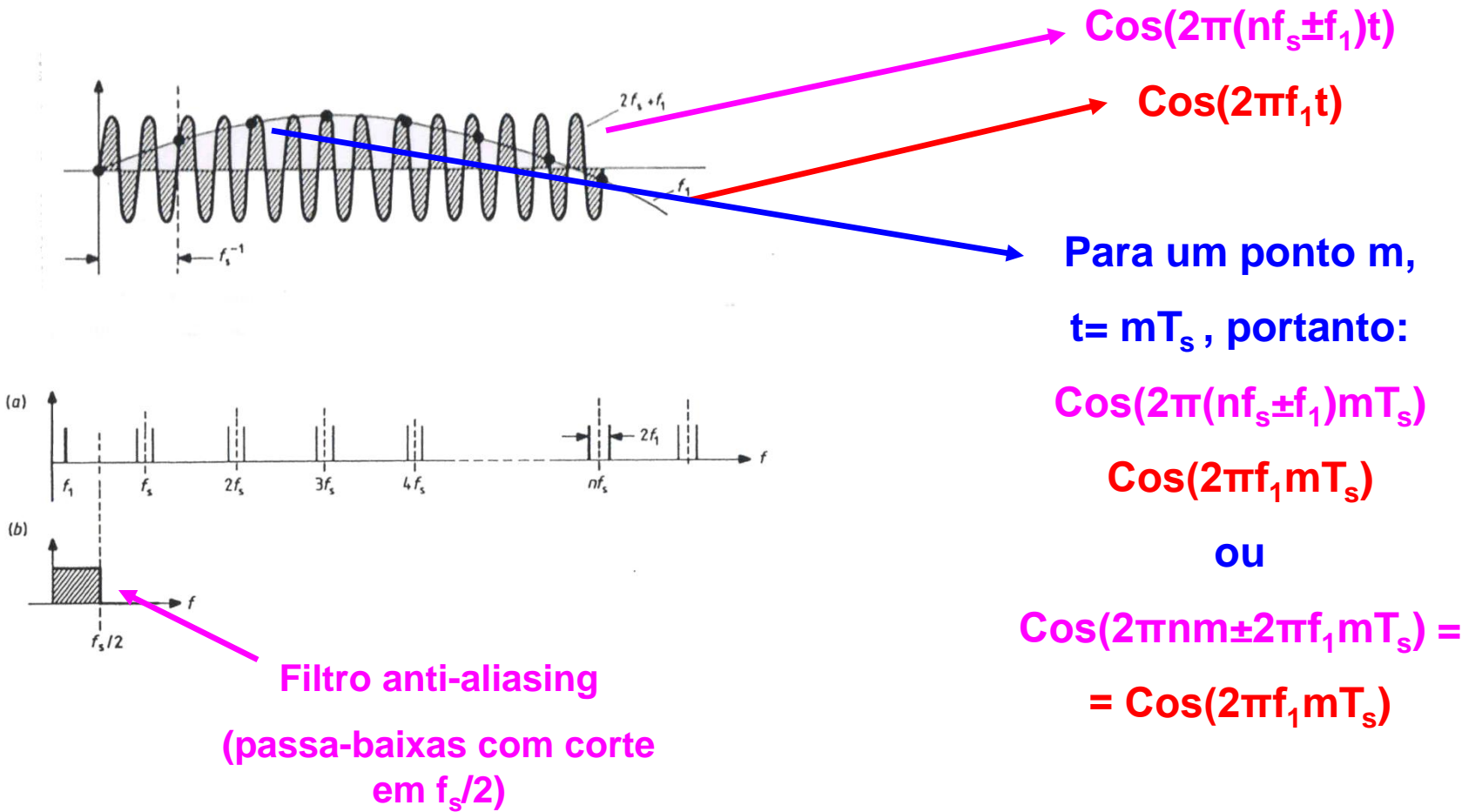
Com o uso de circuitos de “amostragem e sustentação” (track-and-hold) é possível se adquirir múltiplos sinais sincronizadamente, com um único ADC.



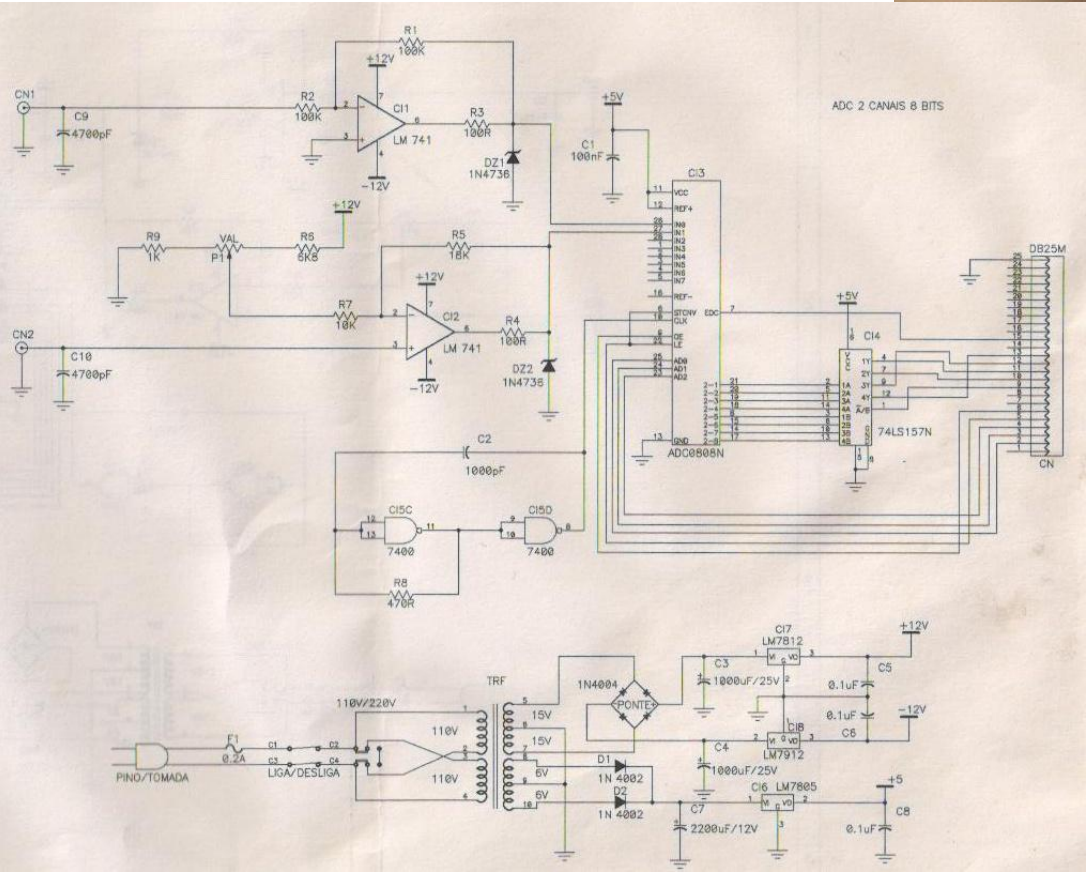
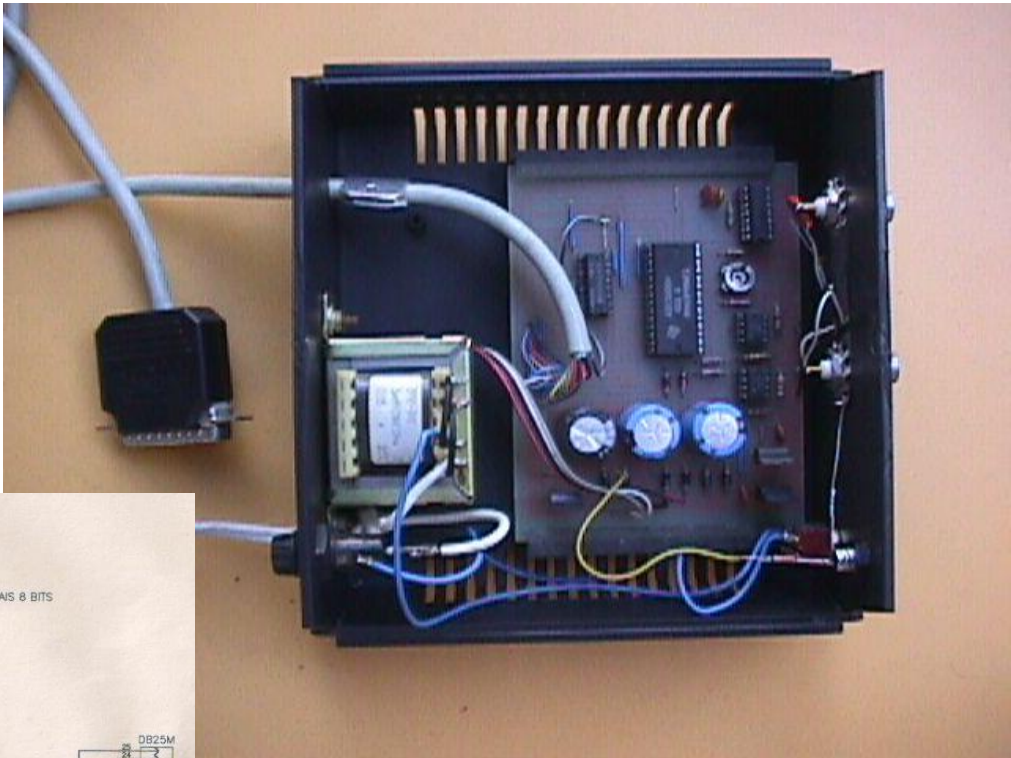
# Ruído e Digitalização

## Conversor Analógico-Digital (ADC)

### Aliasing (Homônimo ?)

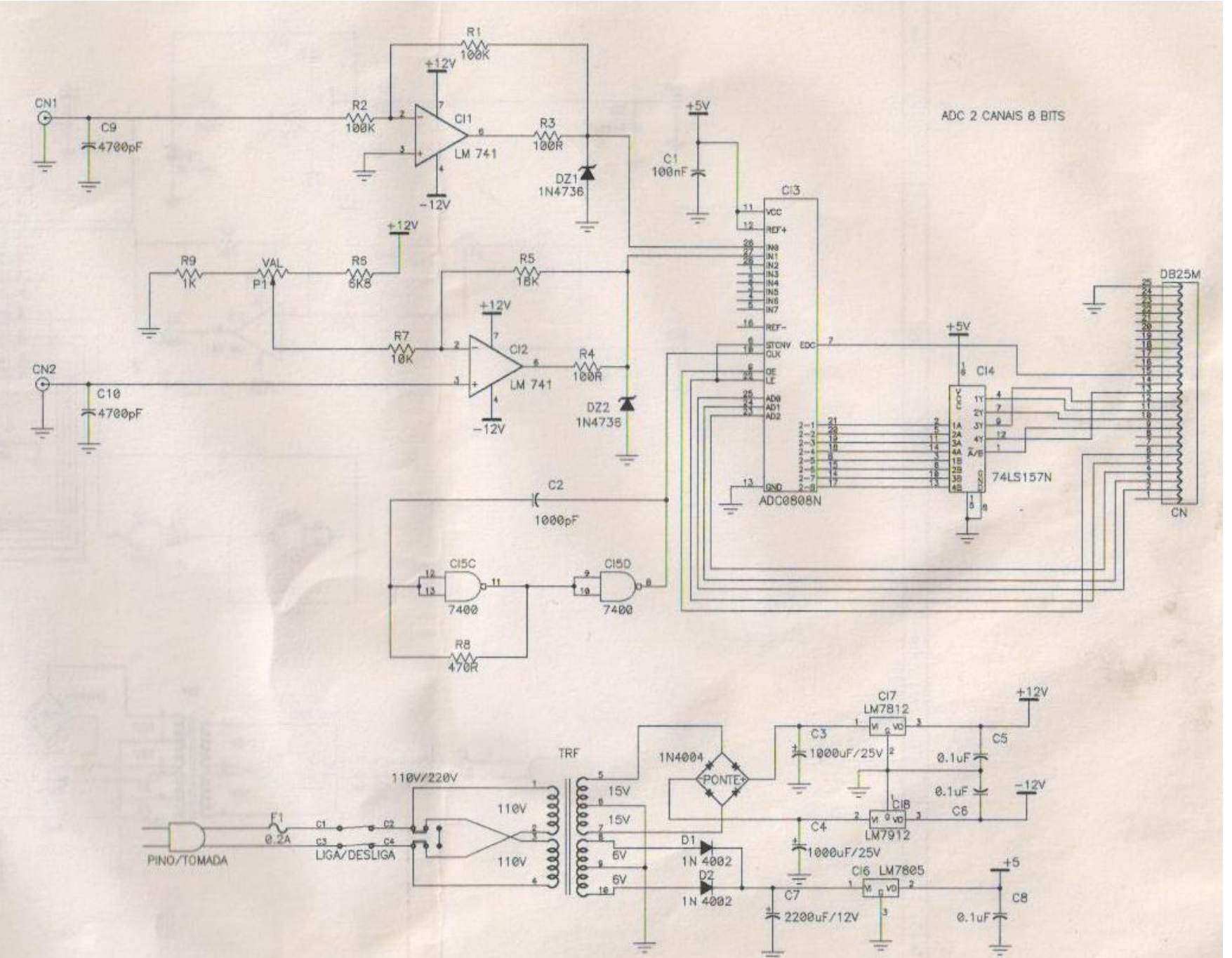


Conversor Analógico-Digital (ADC)



Efeito fotoelétrico  
e Franck-Hertz





# Conversor Analógico-Digital (ADC)

- Total Unadjusted Error . . .  $\pm 0.75$  LSB Max for ADC0808 and  $\pm 1.25$  LSB Max for ADC0809
- Resolution of 8 Bits
- 100  $\mu$ s Conversion Time
- Ratiometric Conversion
- Monotonicity Over the Entire A/D Conversion Range
- No Missing Codes
- Easy Interface with Microprocessors
- Latched 3-State Outputs
- Latched Address Inputs
- Single 5-V Supply
- Low Power Consumption

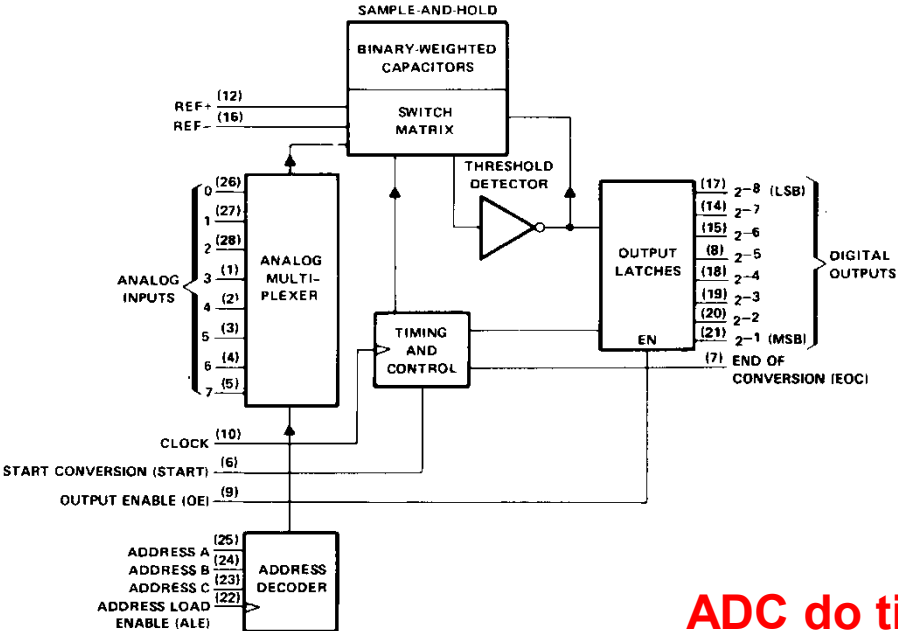
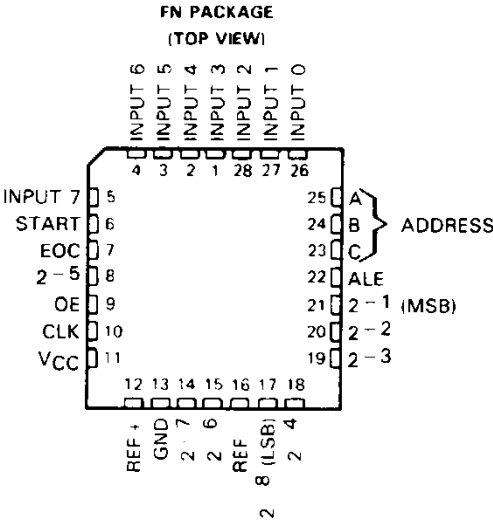
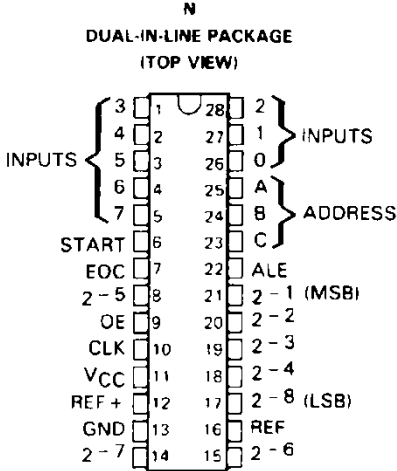
MULTIPLEXER FUNCTION TABLE

INPUTS				SELECTED ANALOG CHANNEL
ADDRESS			ADDRESS STROBE	
C	B	A		
L	L	L	$\uparrow$	0
L	L	H	$\uparrow$	1
L	H	L	$\uparrow$	2
L	H	H	$\uparrow$	3
H	L	L	$\uparrow$	4
H	L	H	$\uparrow$	5
H	H	L	$\uparrow$	6
H	H	H	$\uparrow$	7

H = high level, L = low level  
 $\uparrow$  = low-to-high transition

## ADC0808, ADC0809 CMOS ANALOG-TO-DIGITAL CONVERTERS WITH 8-CHANNEL MULTIPLEXERS

D2642, JUNE 1981—REVISED MAY 1988



ADC do tipo “contador”

# Conversor Analógico-Digital (ADC)

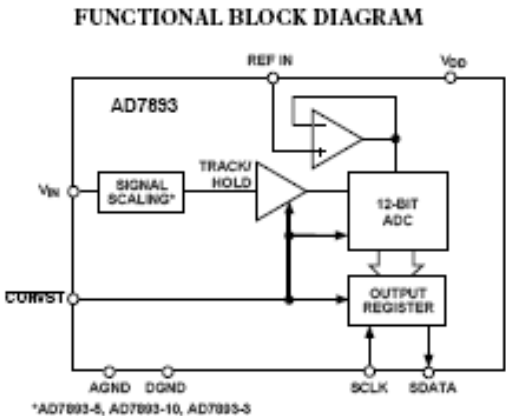


**ANALOG  
DEVICES**

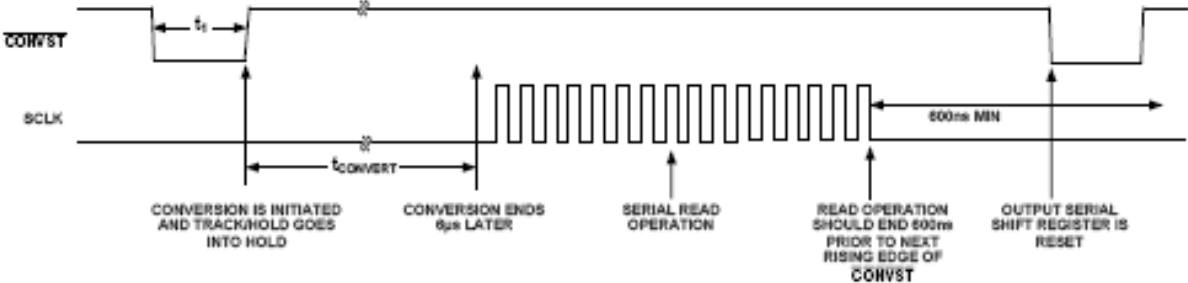
**LC<sup>2</sup>MOS 12-Bit, Serial 6  $\mu$ s  
ADC in 8-Pin Package**

**AD7893**

- FEATURES**
- Fast 12-Bit ADC with 6  $\mu$ s Conversion Time
  - 8-Pin Mini-DIP and SOIC
  - Single Supply Operation
  - High Speed, Easy-to-Use, Serial Interface
  - On-Chip Track/Hold Amplifier
  - Selection of Input Ranges
    - $\pm 10$  V for AD7893-10
    - $\pm 2.5$  V for AD7893-3
    - 0 V to +2.5 V for AD7893-2
    - 0 V to +5 V for AD7893-5
  - Low Power: 25 mW typ



PIN FUNCT		
Pin No.	Pin Mnemonic	Description
1	REF IN	Voltage Reference Input.
2	VIN	Analog Input Channel.
3	AGND	Analog Ground.
4	SCLK	Serial Clock Input..
5	SDATA	Serial Data Output..
6	DGND	Digital Ground.
7	CONVST	Convert Start.
8	VDD	Positive supply voltage, +5 V $\pm$ 5%.



ADC de “dupla rampa”

Microcontroladores (PIC)

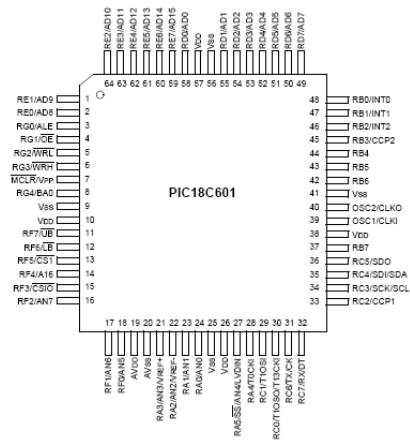
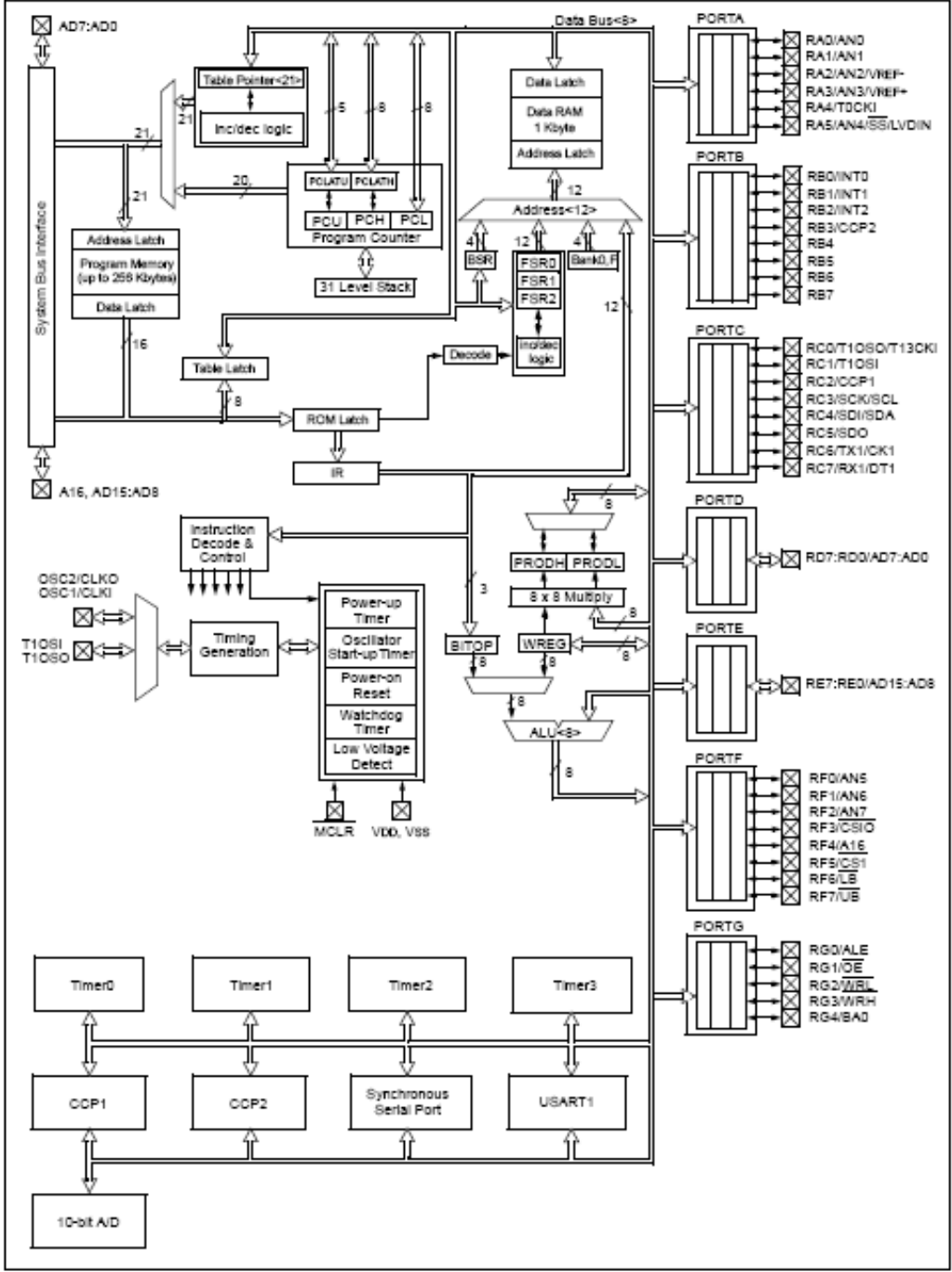
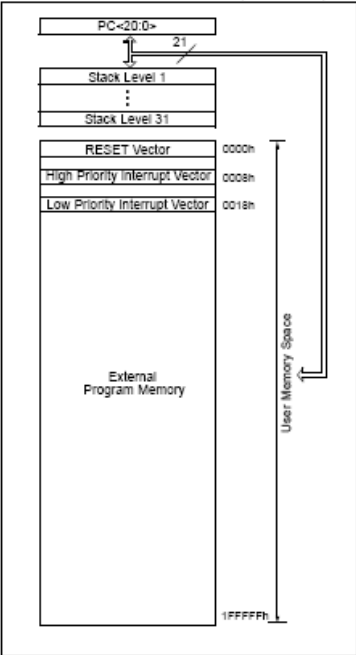


FIGURE 4-2: PROGRAM MEMORY MAP AND STACK FOR PIC18C801 (PGRM = 0)





## Microcontroladores (PIC)

TABLE 1-1: DEVICE FEATURES

Features		PIC18C601	PIC18C801
Operating Frequency		DC - 25 MHz	DC - 25 MHz
External Program Memory	Bytes	256K	2M
	Max. # of Single Word Instructions	128K	1M
Data Memory (Bytes)		1536	1536
Interrupt Sources		15	15
I/O Ports		Ports A - G	Ports A - H, J
Timers		4	4
Capture/Compare/PWM modules		2	2
Serial Communications		MSSP, Addressable USART	MSSP, Addressable USART
10-bit Analog-to-Digital Module		8 input channels	12 input channels
RESETS (and Delays)		POR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)	POR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)
Programmable Low Voltage Detect		Yes	Yes
8-bit External Memory Interface		Yes	Yes
8-bit De-multiplexed External Memory Interface		No	Yes
16-bit External Memory Interfaces		Yes	Yes
On-chip Chip Select Signals		$\overline{CS1}$	$\overline{CS1}$ , $\overline{CS2}$
On-chip I/O Chip Select Signal		Yes	Yes
Instruction Set		75 Instructions	75 Instructions
Packages		64-pin TQFP 68-pin PLCC	80-pin TQFP 84-pin PLCC

## Digital Signal Processor (DSP)



### Specifications

- 16 bits fixed point TMS320VC5402 DSP @ 100 MHz
- USB data throughput: 0.5 Mb/s

### Memory

- 16K words on-chip (DSP) dual-access RAM, mapped in data space and program space
- 64K words external 1-wait-state static RAM, mapped in data space

### Analog I/Os

- Eight 16 bits sigma-delta analog inputs, sampling at up to 80Ks/s (22Ks/s with anti-aliasing filters); Dynamic range  $\approx 10$  V
- Eight 16 bits sigma-delta analog outputs, sampling at up to 80Ks/s (22Ks/s with anti-aliasing filters); Dynamic range  $\approx 2.5$  V

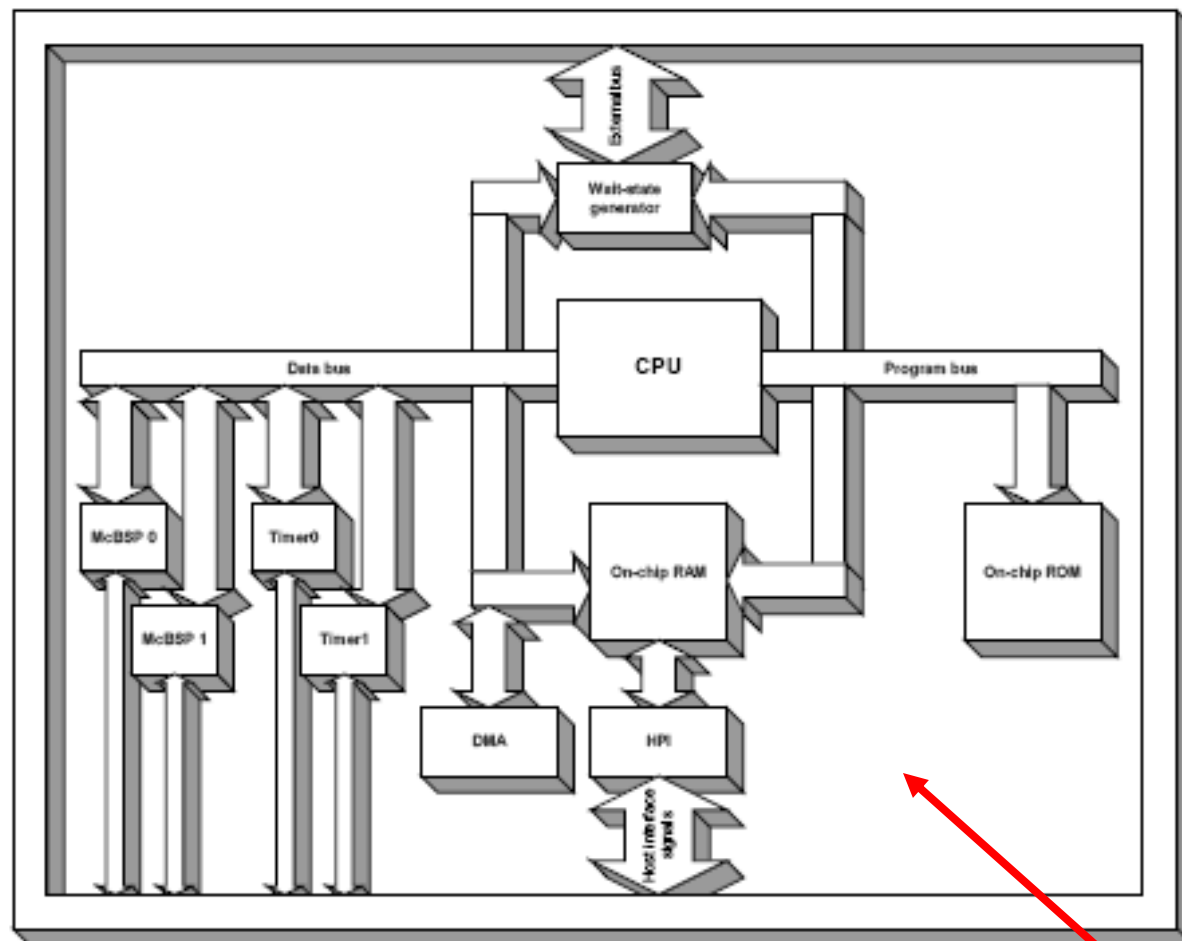


Figure 7-1

Figure 7-1 shows a simplified description of the architecture of the TMS320VC5402

## Digital Signal Processor (DSP)

### Gnome X Scanning Microscopy

The GXSM is the Gnome X Scanning Microscopy project, it is a bit more than just a piece of software (the GXSM itself), there is full hardware support for DSP cards including open source DSP software and a growing set of Scanning Probe Microscopy (SPM) related electronics. And it is not limited to SPM at all, it provides generic multidimensional image and data movie processing.

<http://gxsm.sourceforge.net/>





Figure 2-2

"Bikeman at DSPWorld 1997" Photo: Cory Roy

## Digital Signal Processor (DSP)

The microprocessor used in this application is a TMS320C50 from Texas Instruments.

The system has 3 inputs that are brought to the CPU via 3 peripherals:

- A Pulse Width Modulation (PWM) signal follows the position of a radio-control joystick. The width of the pulses transmitted by the radio-control system represent the trajectory input of the pilot. The signal is transmitted to a peripheral called a *timer* that measures the width of its pulses, and provides this information to the CPU.
- A vibrating gyrometer is placed in the head of the cyclist. It provides an analog signal proportional to the rate of roll of the bicycle around the axis passing through the contact points of the wheels. This analog signal is measured by a peripheral called an Analog to Digital converter that provides this information to the CPU.
- An optical sensor placed in the gear train of the bicycle provides a square wave with a frequency proportional to the speed of the bicycle. This frequency is measured and provided to the CPU by another timer.

The system has one output that is generated by the CPU:

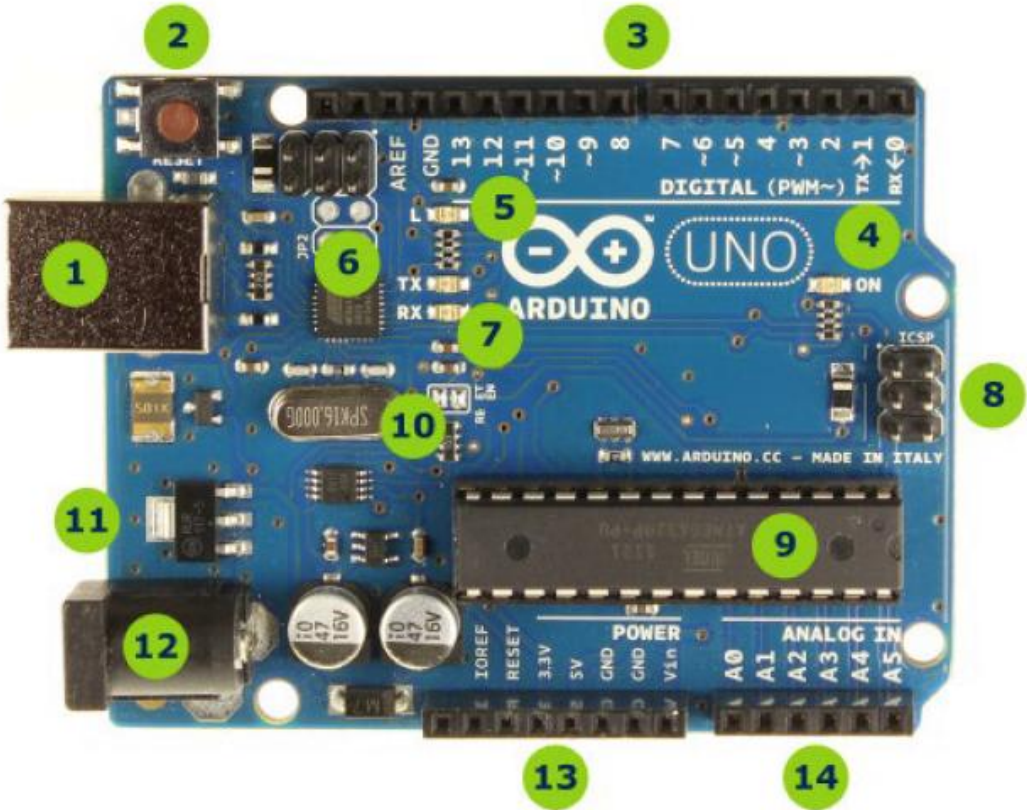
- A PWM signal is generated by a third timer, and sent to the servomotor that controls the position of the handlebars.



# Arduino



Figure 2-2 "Bikeman at DSPWorld 1997" Photo: Cory Roy



- 1 - Conector USB para o cabo tipo AB
- 2 - Botão de reset
- 3 - Pinos de entrada e saída digital e PWM
- 4 - LED verde de placa ligada
- 5 - LED laranja conectado ao pin13
- 6 - ATmega encarregado da comunicação com o computador
- 7 - LED TX (transmissor) e RX (receptor) da comunicação serial
- 8 - Porta ICSP para programação serial
- 9 - Microcontrolador ATmega 328, cérebro do Arduino
- 10 - Cristal de quartzo 16Mhz
- 11 - Regulador de voltagem
- 12 - Conector fêmea 2,1mm com centro positivo
- 13 - Pinos de voltagem e terra
- 14 - Entradas analógicas

Arduino/Genuino Uno Board Anatomy

Arduino/Genuino boards senses the environment by receiving inputs from many sensors, and affects their surroundings by controlling lights, motors, and other actuators. Arduino/Genuino boards are the microcontroller development platform that will be at the heart of your projects. When making something you will be building the circuits and interfaces for interaction, and telling the microcontroller how to interface with other components. Here the anatomy of Arduino/Genuino Uno.

- 1. **ATmega microcontroller** The heart of your board.
- 2. **Pin 13 LED** The only actuator built-in to your board. Besides being a handy target for your first blink sketch, this LED is very useful for debugging.
- 3. **Power LED** Indicates that your Genuino is receiving power. Useful for debugging.
- 4. **Digital pins** Use these pins with digitalRead(), digitalWrite(), and analogWrite(). analogWrite() works only on the pins with the PWM symbol.
- 5. **Analog in** Use these pins with analogRead().
- 6. **GND and 5V pins** Use these pins to provide +5V power and ground to your circuits.
- 7. **Power connector** This is how you power your Genuino when it's not plugged into a USB port for power. Can accept voltages between 7-12V.
- 8. **TX and RX LEDs** These LEDs indicate communication between your Genuino and your computer. Expect them to flicker rapidly during sketch upload as well as during serial communication. Useful for debugging.
- 9. **USB port** Used for powering your Genuino Uno, uploading your sketches to your Genuino, and for communicating with your Genuino sketch (via Serial. println() etc.).
- 10. **Reset button** Resets the ATmega microcontroller.



- 1 - Conector USB para o cabo tipo AB
- 2 - Botão de reset
- 3 - Pinos de entrada e saída digital e PWM
- 4 - LED verde de placa ligada
- 5 - LED laranja conectado ao pin13
- 6 - ATmega encarregado da comunicação com o computador
- 7 - LED TX (transmissor) e RX (receptor) da comunicação serial
- 8 - Porta ICSP para programação serial
- 9 - Microcontrolador ATmega 328, cérebro do Arduino
- 10 - Cristal de quartzo 16Mhz
- 11 - Regulador de voltagem
- 12 - Conector fêmea 2,1mm com centro positivo
- 13 - Pinos de voltagem e terra
- 14 - Entradas analógicas



# Família Arduino

Arduino Leonardo



Arduino Mega2560 R3



Arduino Esplora



Arduino Mega ADK



Arduino Pro



Arduino FIO V3



Arduino LilyPad



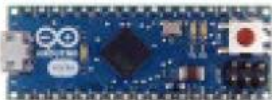
Arduino Mini 05



Arduino Pro Mini



Arduino Micro



## Shields p/ Arduino

Arduino Ethernet Shield R3



Kit Motor Shield R3



Arduino WiFi Shield



Arduino XBee Shield



Arduino ProtoShield R3



Kit Joystick Shield



Shield celular com SM5100B



Shield GPS



Shield LCD Colorido



Shield MP3 Player



Shield WiFly



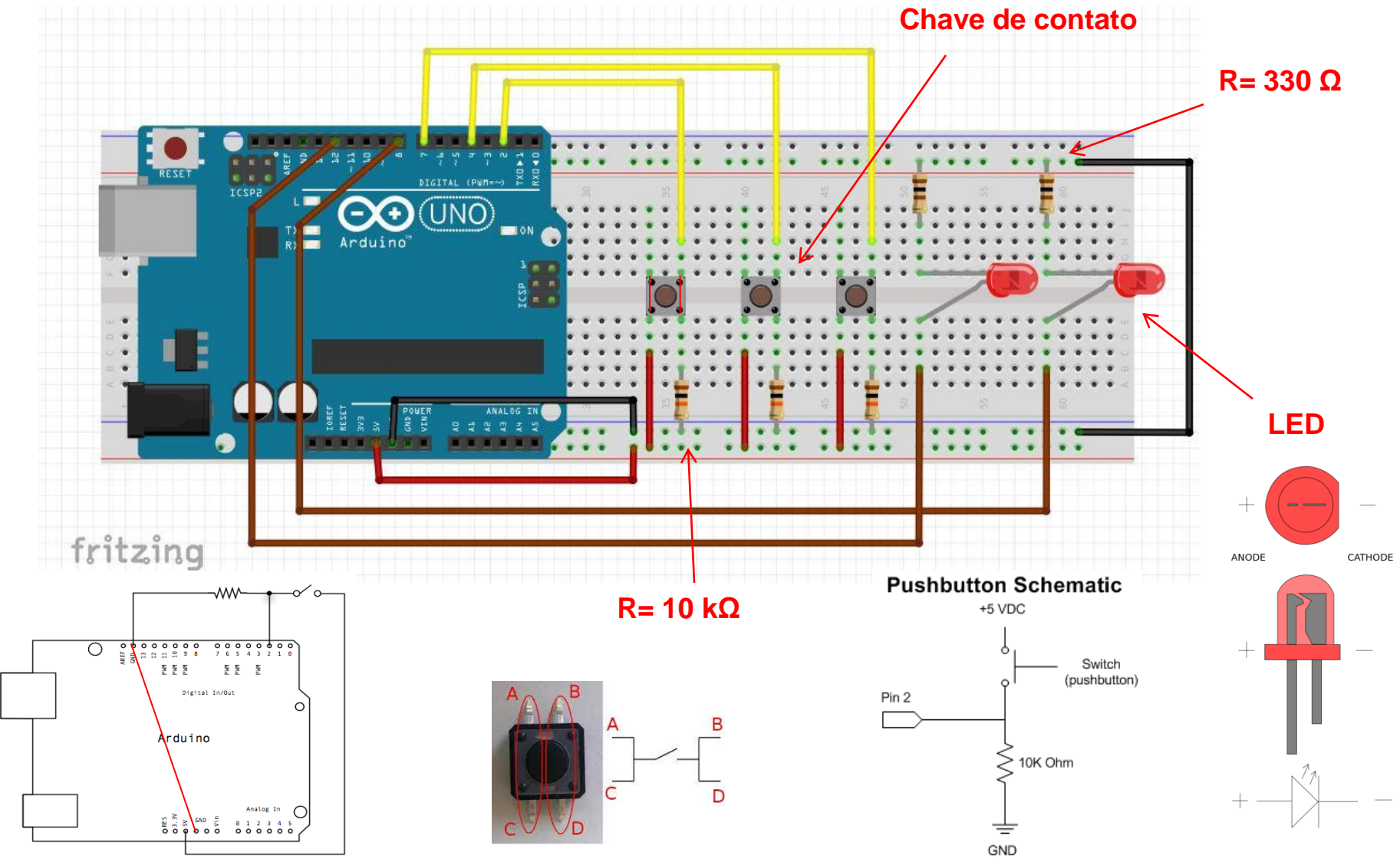
Wireless SD Shield





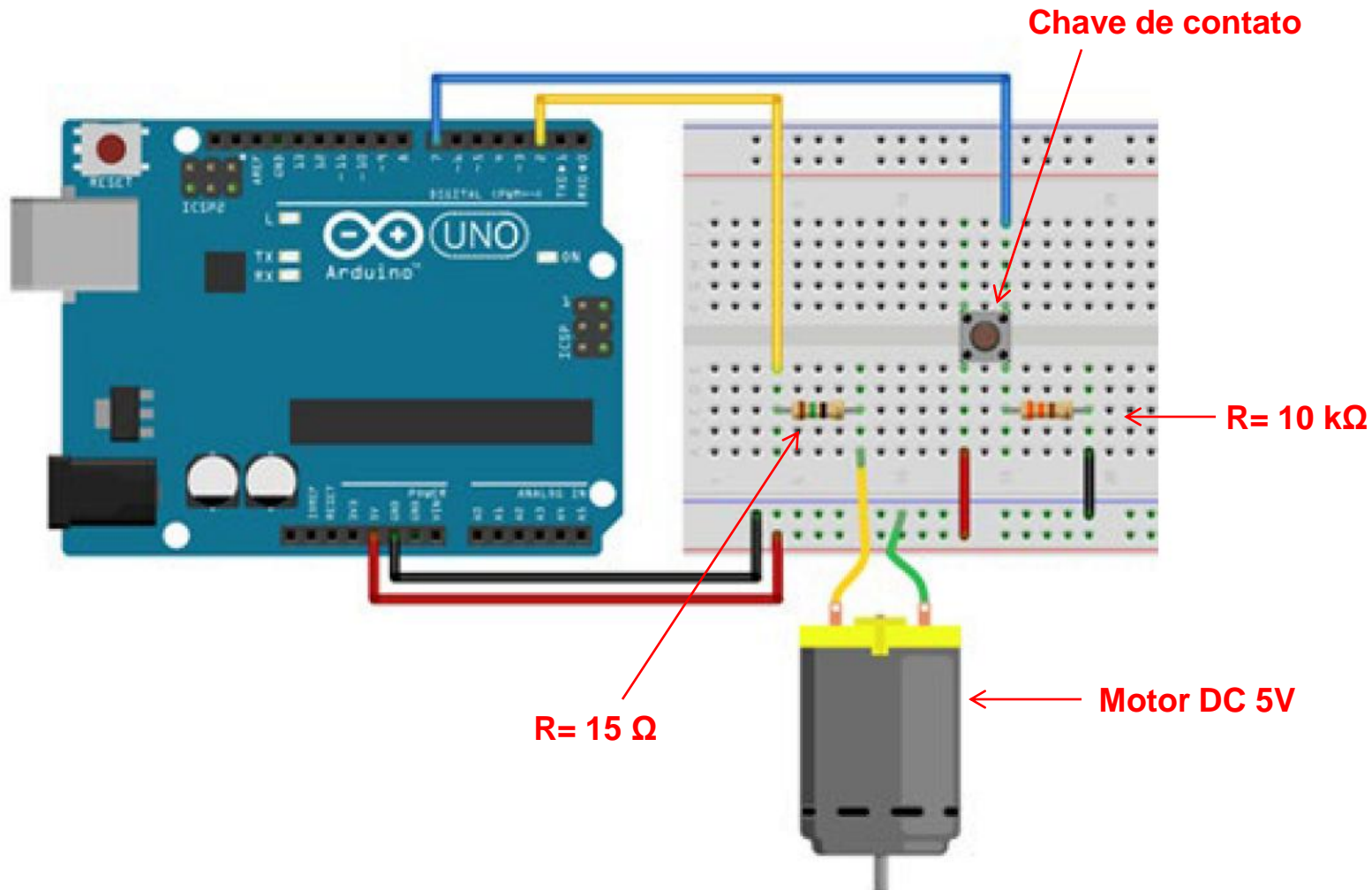
# Arduino

## Exercício – Teste de tempo de resposta visual



# Arduino

## Exercício – Acionamento de um motor de corrente contínua



# Arduino

[Buy](#)[Download](#)[Products](#)[Learning](#)[Forum](#)[Support](#)[Blog](#)[LOG IN](#)

## Structure

- `setup()`
- `loop()`

### Control Structures

- `if`
- `if...else`
- `for`
- `switch case`
- `while`
- `do... while`
- `break`
- `continue`
- `return`
- `goto`

### Further Syntax

## Variables

### Constants

- `HIGH` | `LOW`
- `INPUT` | `OUTPUT` | `INPUT_PULLUP`
- `LED_BUILTIN`
- `true` | `false`
- integer constants
- floating point constants

### Data Types

- `void`
- `boolean`
- `char`
- `unsigned char`
- `byte`
- `int`
- `unsigned int`

## Functions

### Digital I/O

- `pinMode()`
- `digitalWrite()`
- `digitalRead()`

### Analog I/O

- `analogReference()`
- `analogRead()`
- `analogWrite()` - *PWM*

### Due only

- `analogReadResolution()`
- `analogWriteResolution()`

### Advanced I/O

- `tone()`

### 3) Elaboração dos Projetos

- **Título e Proponentes**

- **Resumo**

- **Introdução e motivação**

(relevância e estado da arte no tema, vantagens/desvantagens, ...

( O que fazer? Porquê fazer? Como fazer? ) )

- **Justificativa tecnico-científica**

(facilidade de operação, melhores resultados, flexibilidade, ...)

- **Detalhamento do projeto**

- **Orçamento**

**Texto com 1 página.**