

Universidade Federal do Rio Grande do Norte  
Departamento de Engenharia de Computação e  
Automação

**Sistemas Embarcados:  
Interfaces com o Mundo Analógico**

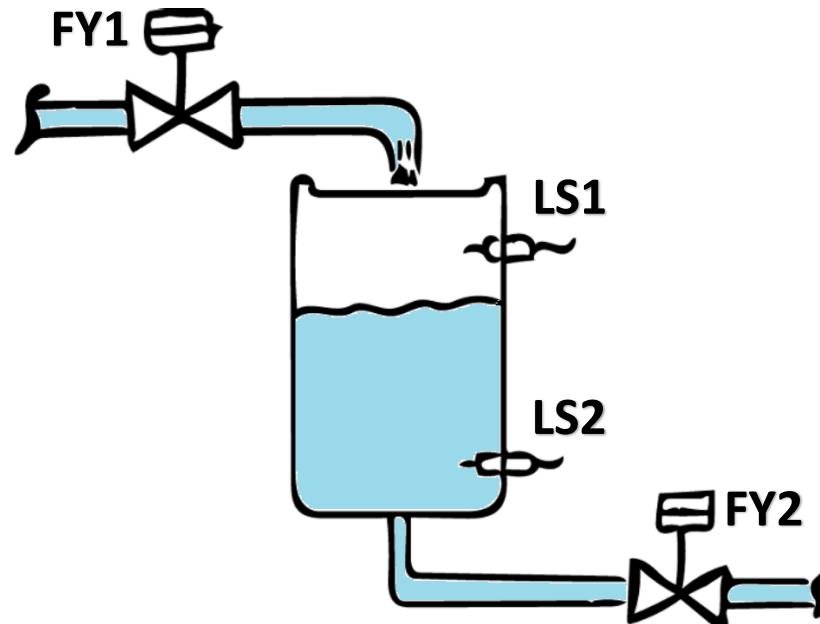
DCA0119 – Sistemas Digitais

Heitor Medeiros Florencio

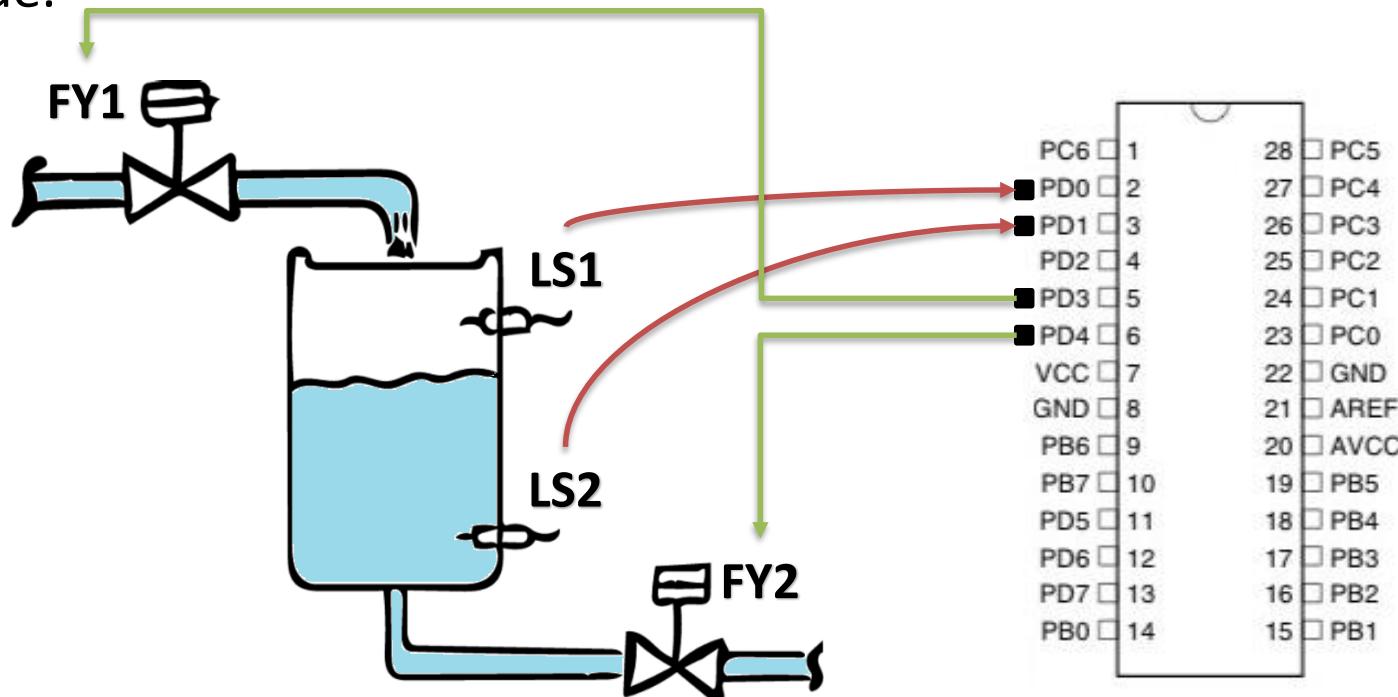
# Tópicos

- Sistemas Embarcados: Iteração com o mundo analógico.
- Conversores AD e DA.
- Conversores AD em Microcontroladores.
- Conversores DA em Microcontroladores.
- PWM.

- **Problema 01:** Projetar um sistema embarcado para controlar o nível de um tanque de água a partir da leitura de duas chaves de nível, **LS1 (chave de nível alto)** e **LS2 (chave de nível baixo)**. O processador deve considerar os sinais digitais dessas chaves para ligar ou desligar duas válvulas, **FY1 (válvula de entrada)** e **FY2 (válvula de saída)**.

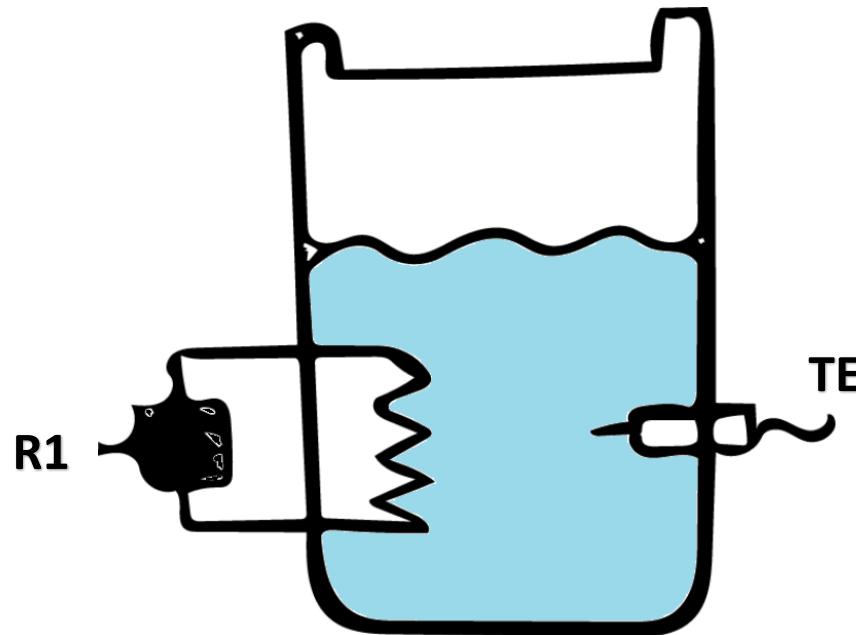


- Solução: Utilizar um microcontrolador para desenvolver a lógica de abertura/fechamento das válvulas a partir do nível de água no tanque.

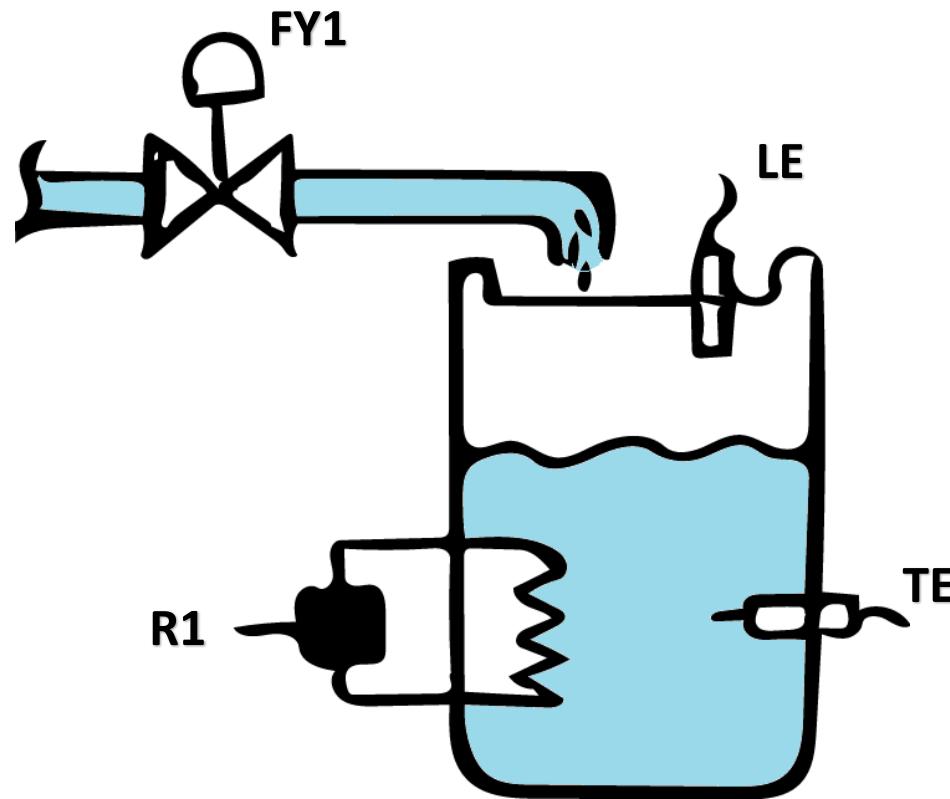


- Entradas digitais: **PD0 e PD1**.
- Saídas digitais: **PD3 e PD4**.

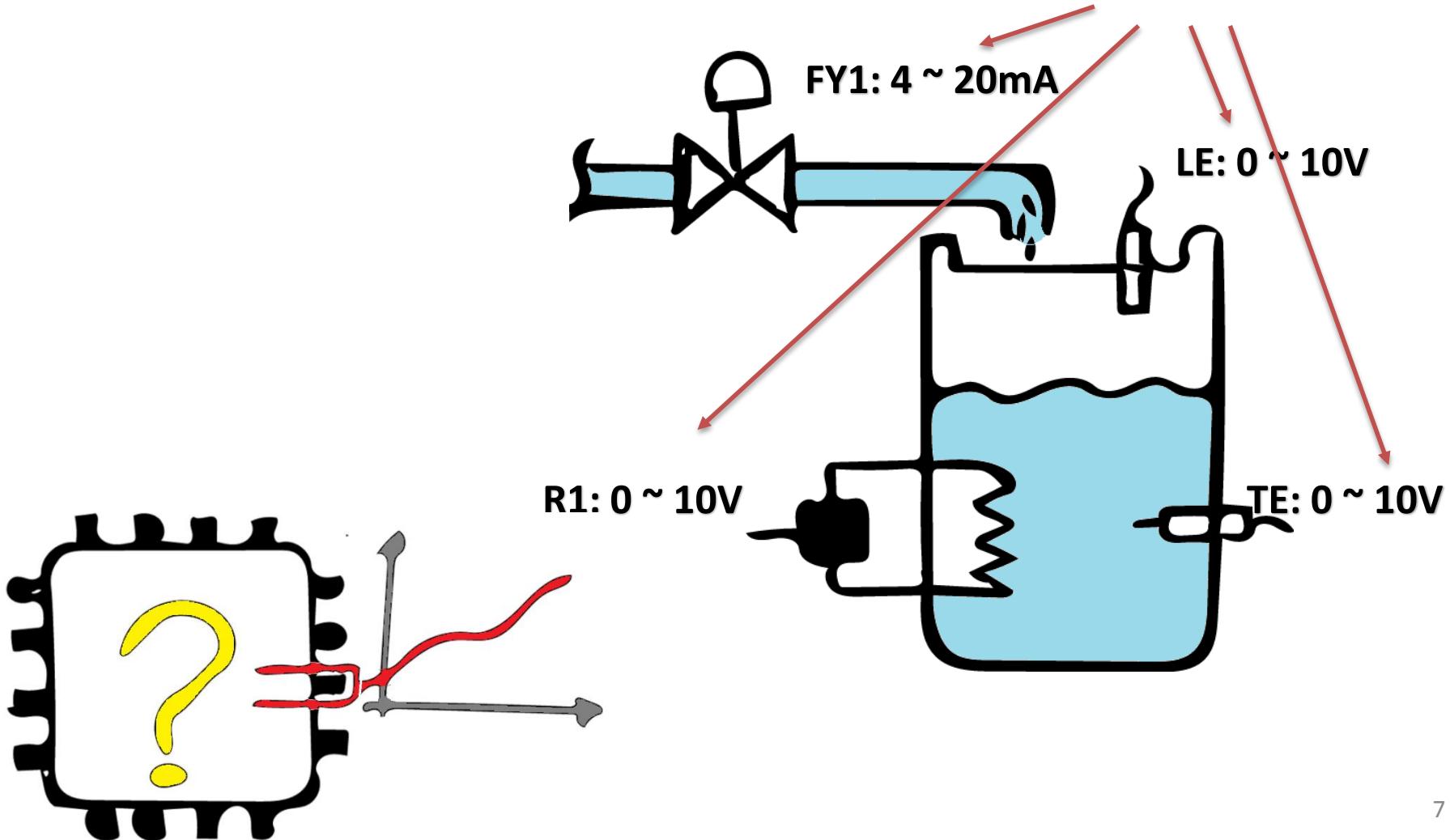
- **Problema 02:** Projetar um sistema embarcado para controlar a temperatura da água de um tanque a partir da leitura de temperatura do sensor, **TE (sensor de temperatura)**. O processador deve considerar o sinal do sensor para controlar a tensão do aquecedor (resistência), **R1 (aquecedor)**.



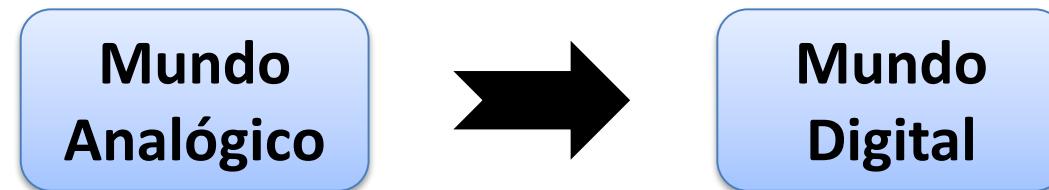
- **Problema 03:** Utilizar um microcontrolador para controlar nível e temperatura.



- Problema: **Sensores e atuadores com variáveis analógicas:**



- Os sistemas de monitoramento e automação de processos/ambientes dependem cada vez mais das tecnologias de sensoriamento e atuação.
- As **grandezas físicas do mundo real são, predominantemente, valores analógicos.**
- Os transdutores, que trabalham em conjunto com os sensores, são responsáveis por transformar/adequar os valores lidos para a unidade de processamento ou de transmissão.

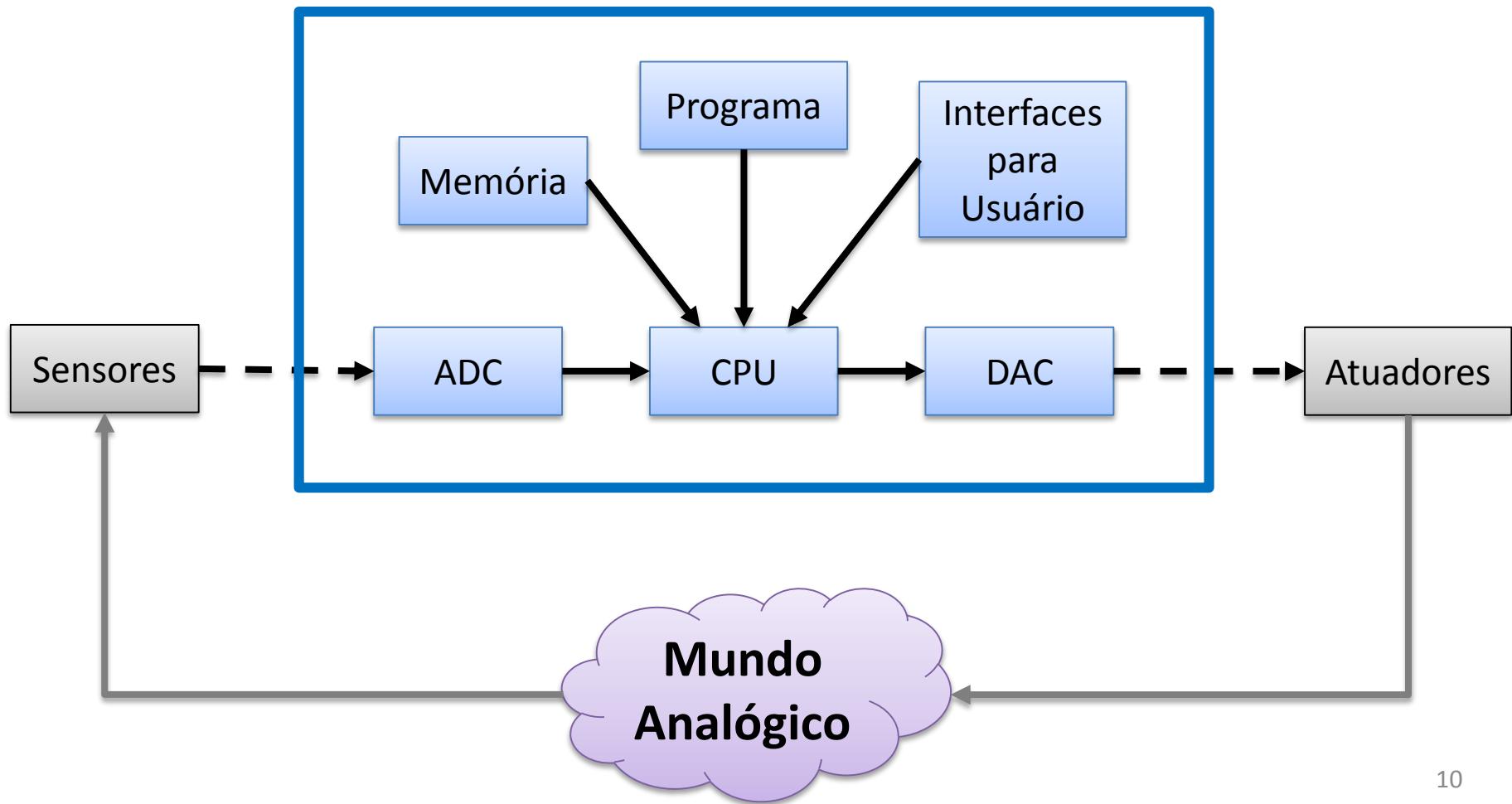


- Grandezas físicas analógicas:

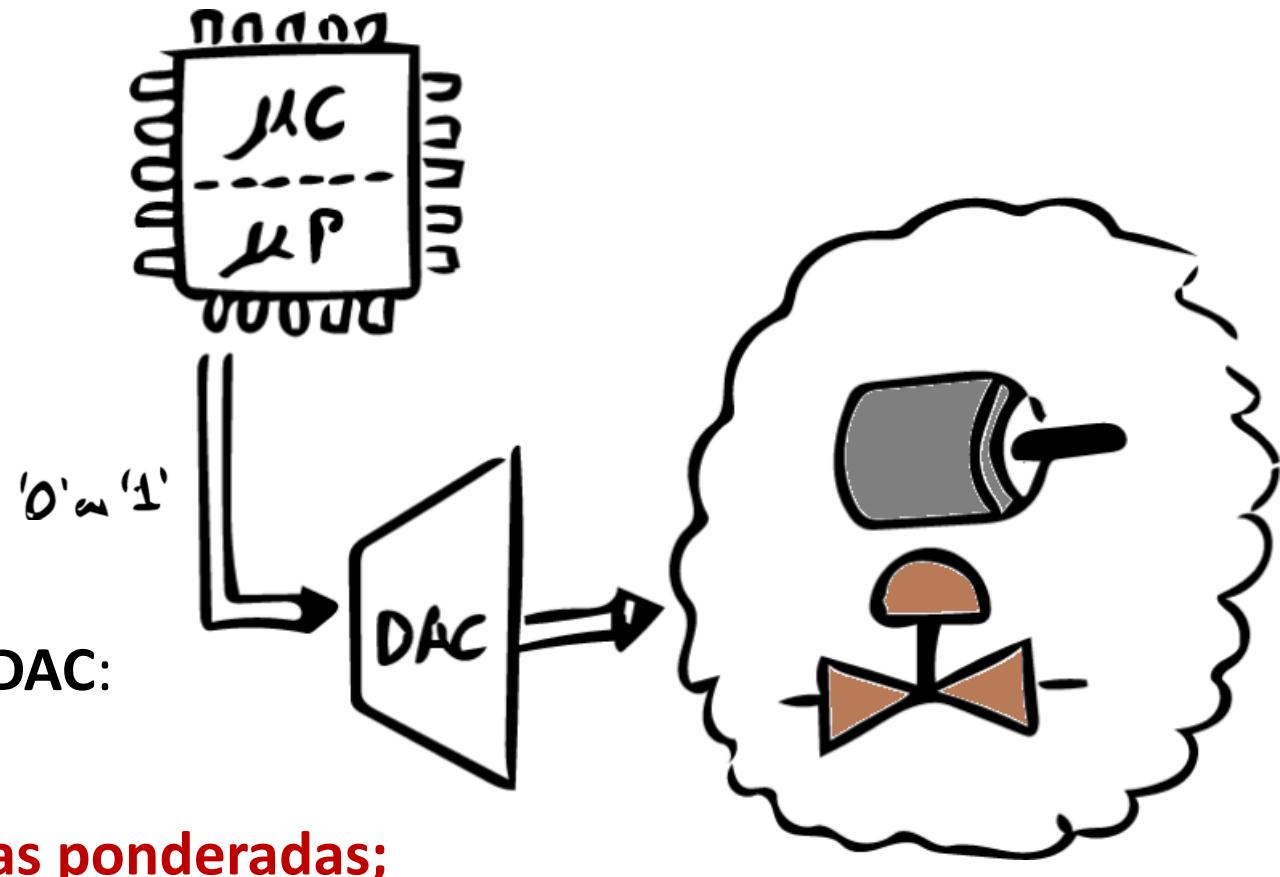
- Nível;
- Temperatura;
- Pressão;
- Posição;
- Luminosidade;
- PH;
- Comprimento;



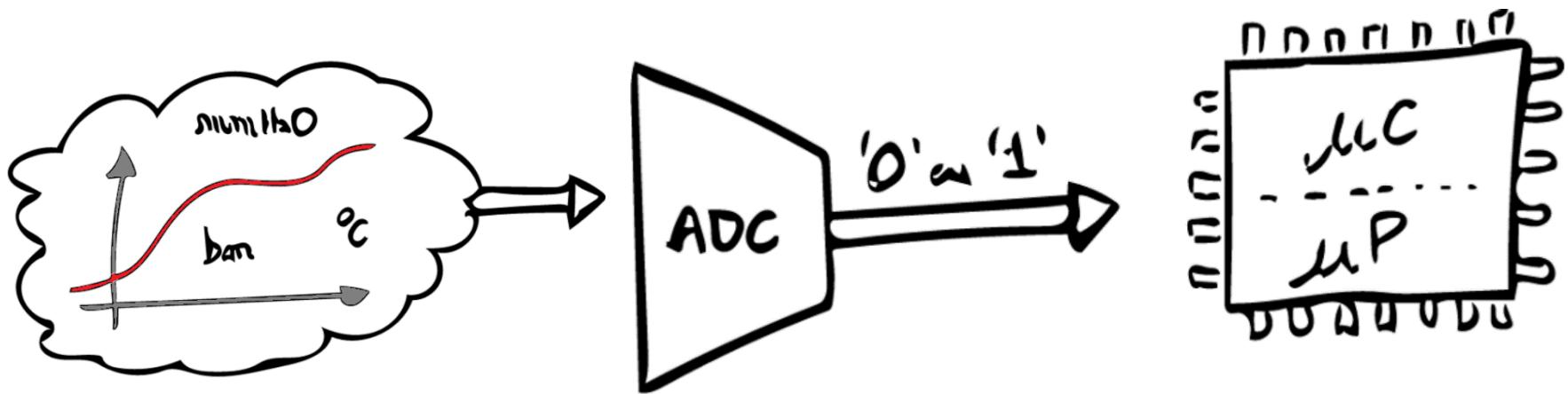
## Sistema Embarcado



- Existem diversos circuitos de conversão de um sinal analógico em digital, ou vice e versa.

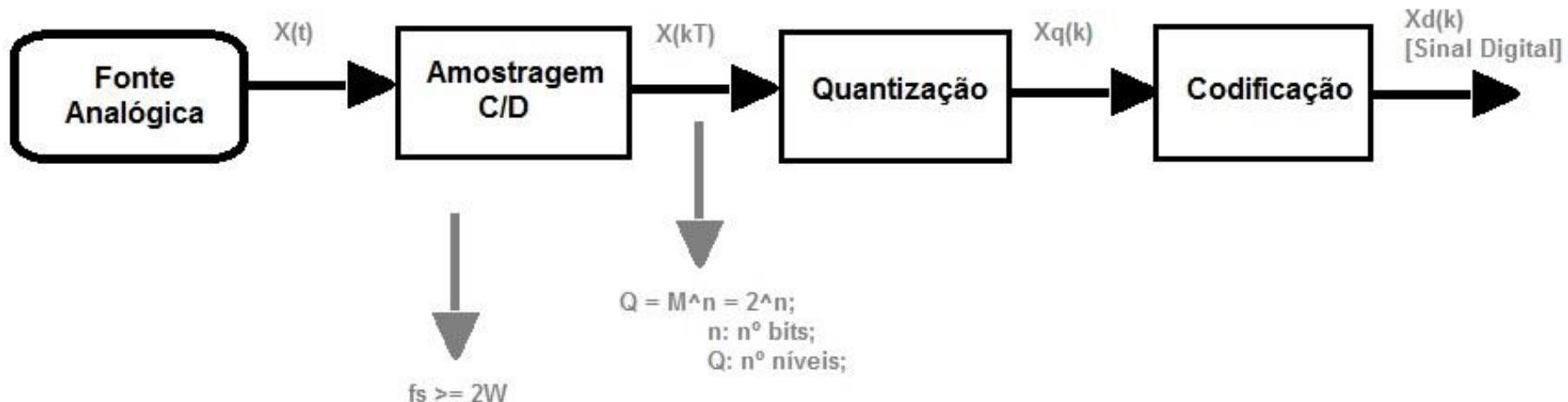


- A conversão AD é mais complexa e, geralmente, consome mais tempo de processamento.

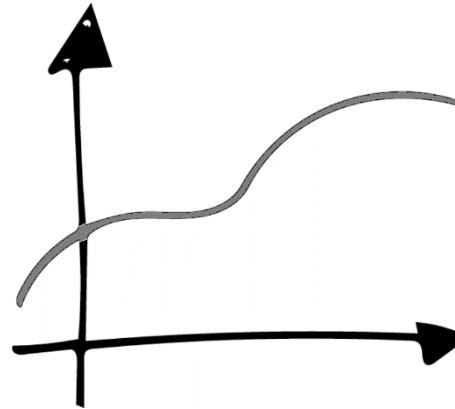


- Exemplos ADC:
  - Rampa;
  - Aproximações sucessivas;
  - Flash;
  - Sigma delta;

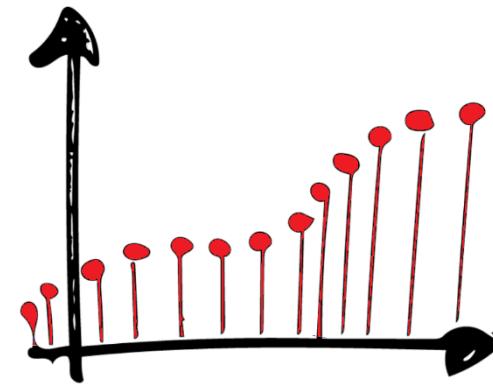
- ADC é um circuito que converte um **sinal analógico**, contínuo no tempo, em um sinal **amostrado no tempo**, **quantizado dentro de um conjunto de valores inteiros** que por sua vez são **codificados em valores digitais**, tornando o sinal digital.



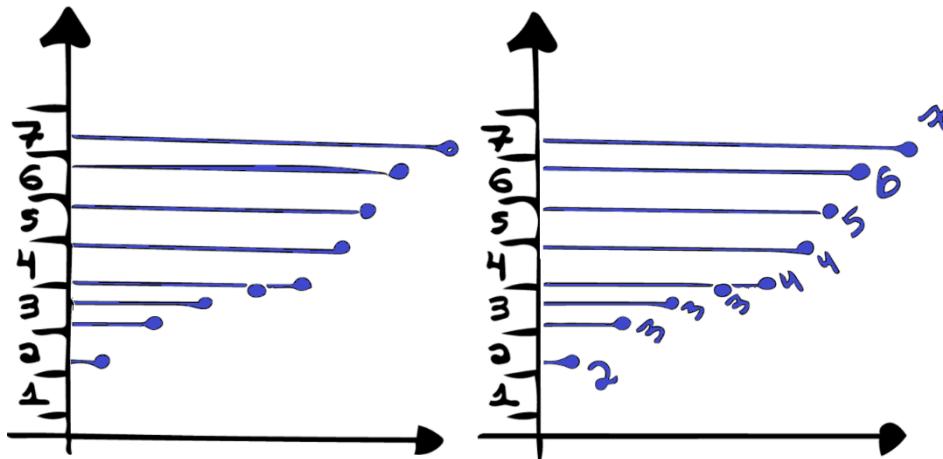
sinal analógico



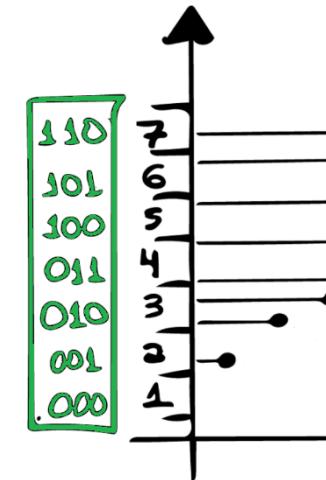
amostrado no tempo



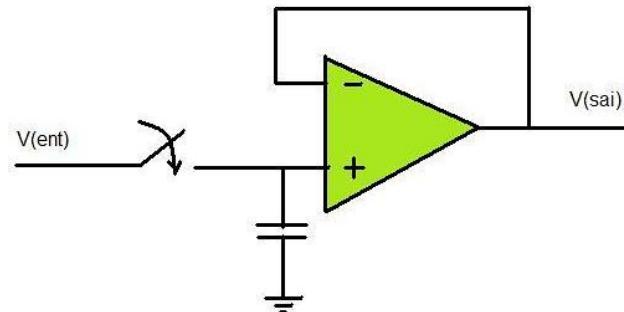
quantizado dentro de um  
conjunto de valores inteiros



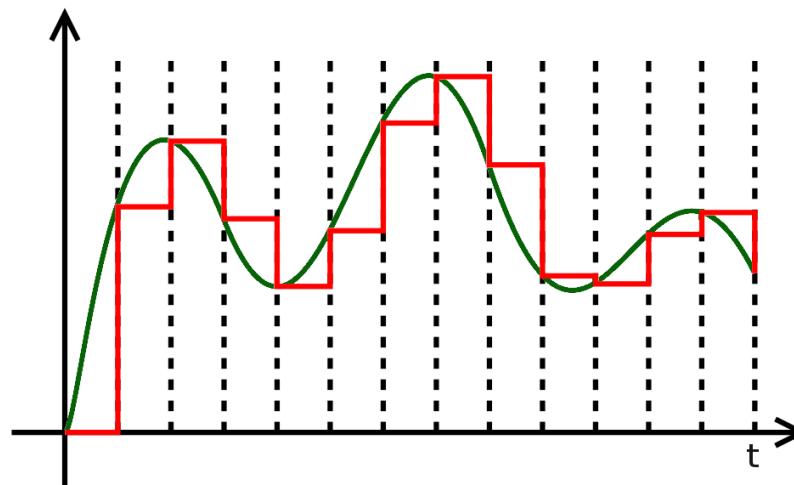
codificados em  
valores digitais



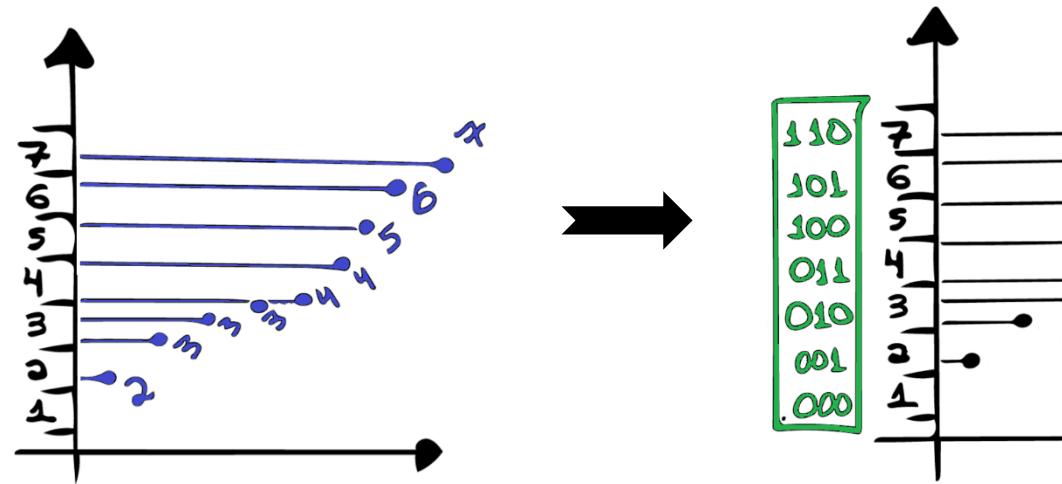
- O circuito mais utilizado para amostragem em microcontroladores é o *sample and hold*.



- Sinal amostrado:



- A precisão da conversão da faixa de quantização (resolução) que por sua vez influencia no número de bits.



- Exemplo:** Um conversor de 10 bits consegue quantizar valores entre 0 e 1023 ( $2^{10}$ ). Se a faixa do valor analógico for **5V**, a resolução da conversão é **4,88 mV**.

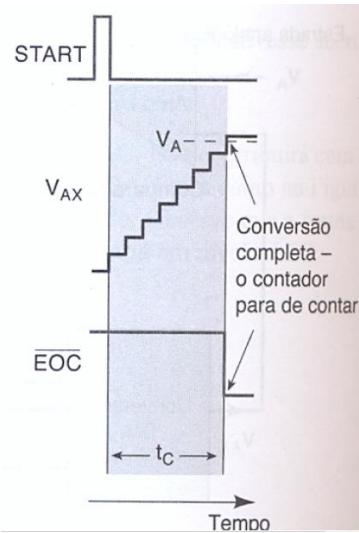
- Taxa de amostragem;
- Número de bits e resolução;
  - O número de bits utilizado na representação do sinal, junto a faixa da variável analógica, determina a resolução do conversor.
- Relação sinal/ruído;
- Tempo de conversão;
- Precisão;
- Erro de offset (DACs);
  - Saída do DAC deveria ser 0 (V ou A) para entrada de bits em '0'.
- Necessidade de se utilizar um filtro de anti-aliasing;
  - Teorema de Nyquist: freq de amostragem, pelo menos, 2x freq do sinal.



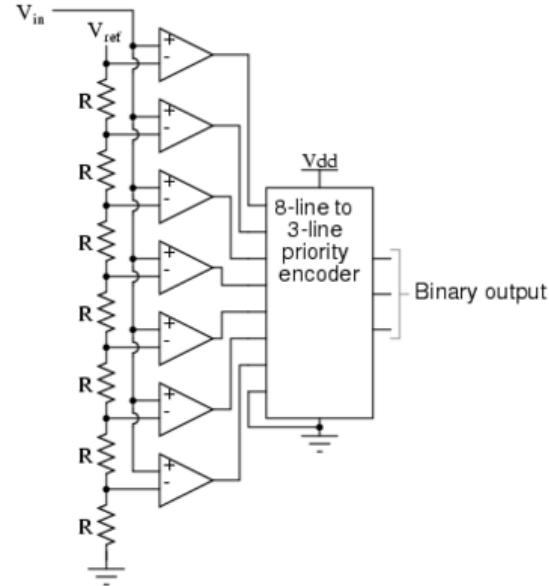
**Nem sempre é possível (ex.: sinais com frequência muito alta)**

# Alguns ADC

- ADC rampa;
  - Contador verificando se  $V_A > V_{AX}$ .



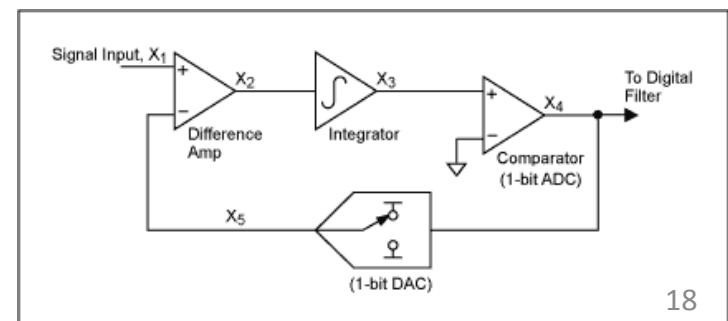
- ADC flash;



- ADC de aproximações sucessivas;



- ADC sigma delta;



## PIC16F87X

### 11.0 ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital (A/D) Converter module has five inputs for the 28-pin devices and eight for the other devices.

The analog input charges a sample and hold capacitor. The output of the sample and hold capacitor is the input into the converter. The converter then generates a digital result of this analog level via successive approximation. The A/D conversion of the analog input signal results in a corresponding 10-bit digital number. The A/D module has high and low voltage reference input that is software selectable to some combination of VDD, Vss, RA2, or RA3.

The A/D converter has a unique feature of being able

## PIC18F2XK20/4XK20

### 19.0 ANALOG-TO-DIGITAL CONVERTER (ADC) MODULE

The Analog-to-Digital Converter (ADC) allows conversion of an analog input signal to a 10-bit binary representation of that signal. This device uses analog inputs, which are multiplexed into a single sample and hold circuit. The output of the sample and hold is connected to the input of the converter. The converter generates a 10-bit binary result via successive approximation and stores the conversion result into the

**ATmega48PA/88PA/168PA/328P**

### 23. Analog-to-Digital Converter

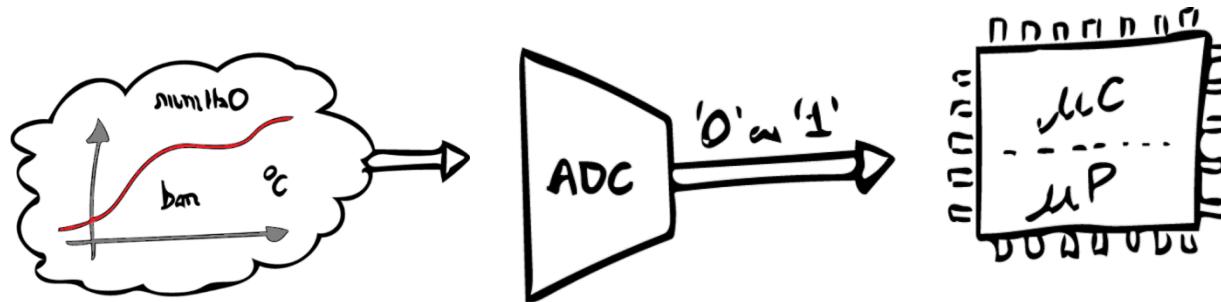
#### 23.1 Features

- 10-bit Resolution
- 0.5 LSB Integral Non-linearity
- $\pm 2$  LSB Absolute Accuracy
- 13 - 260  $\mu$ s Conversion Time
- Up to 76.9 kSPS (Up to 15 kSPS at Maximum Resolution)
- 6 Multiplexed Single Ended Input Channels
- 2 Additional Multiplexed Single Ended Input Channels (TQFP and QFN/MLF Package only)
- Temperature Sensor Input Channel
- Optional Left Adjustment for ADC Result Readout
- 0 -  $V_{CC}$  ADC Input Voltage Range
- Selectable 1.1V ADC Reference Voltage
- Free Running or Single Conversion Mode
- Interrupt on ADC Conversion Complete
- Sleep Mode Noise Canceler

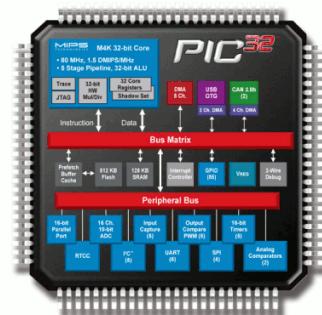
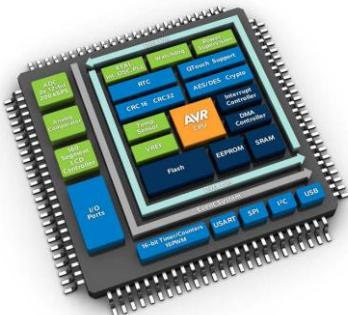
#### 23.2 Overview

The ATmega48PA/88PA/168PA/328P features a 10-bit successive approximation ADC. The ADC is connected to an 8-channel Analog Multiplexer which allows eight single-ended voltage inputs constructed from the pins of Port A. The single-ended voltage inputs refer to 0V (GND).

The ADC contains a Sample and Hold circuit which ensures that the input voltage to the ADC is held at a constant level during conversion. A block diagram of the ADC is shown in Figure 23-1 on page 251.



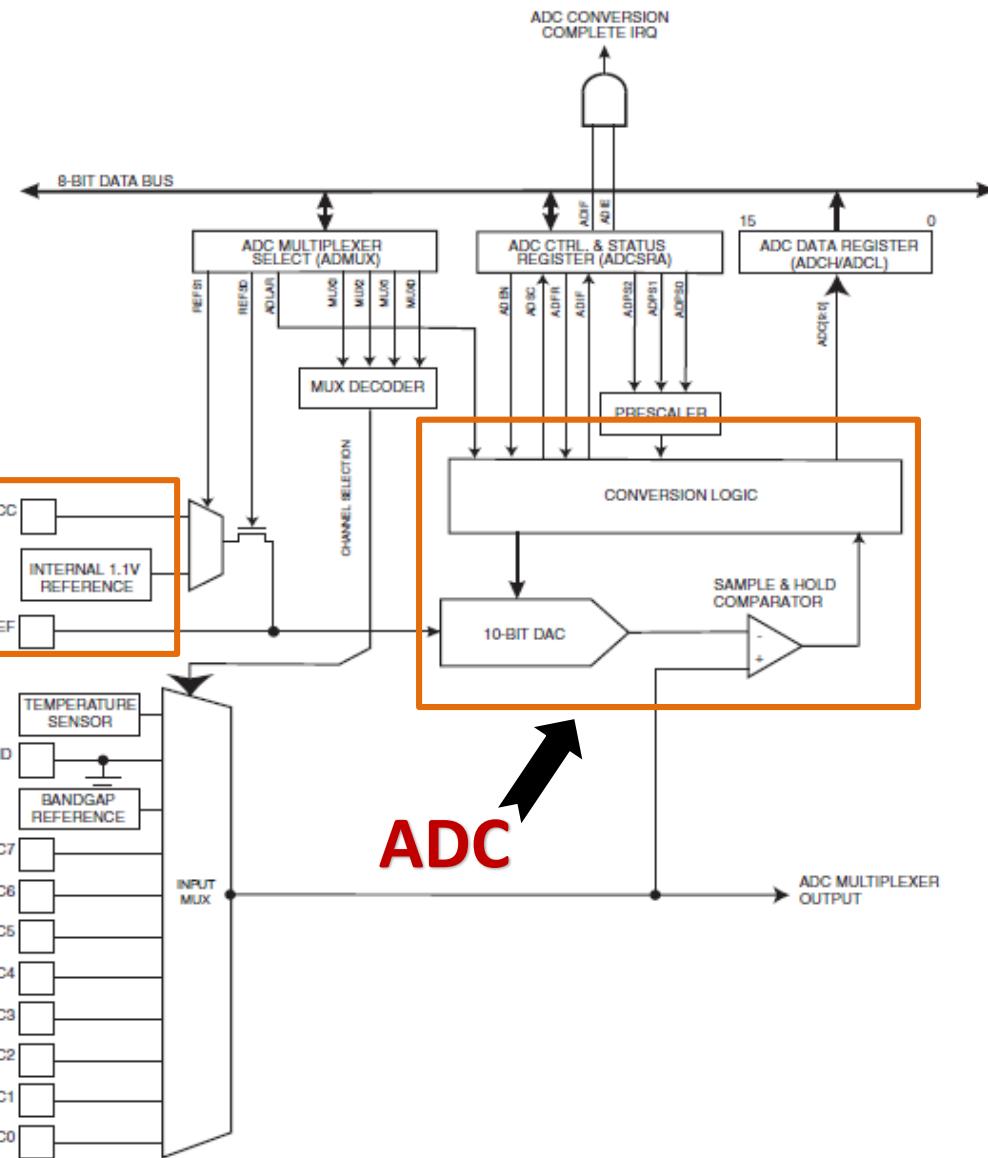
# Conversão AD em Microcontroladores



- Conversor A/D de aproximações sucessivas;
- Resolução de 10 bits;
- 6 canais (ADC0, ADC1, ADC2, ADC3, ADC4, ADC5);
- Faixa de tensão para conversão:
  - Referência interna 1.1V;
  - Referência VCC;
  - Referência externa (máximo Vcc).

(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

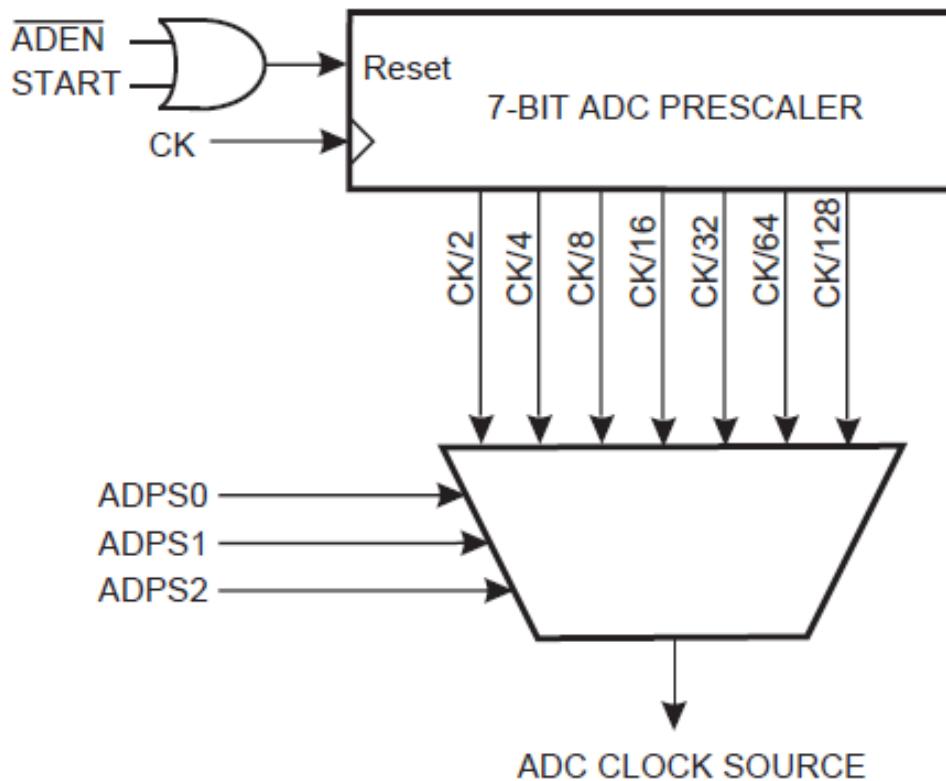
- Esquemático:



**ADC**

**Referência** →  
**Canais** →

- A amostragem depende diretamente do clock do ADC.
- O clock do ADC é uma fração do clock principal.
- O clock pode assumir vários valores:



- Os bits **ADPS** do registrador **ADCSRA** (ADC Control and Status Register A) é responsável por selecionar o clock.

ADCSRA – ADC Control and Status Register A

Bit (0x7A)	7	6	5	4	3	2	1	0	
Read/Write	R/W	ADCSRA							
Initial Value	0	0	0	0	0	0	0	0	

- Bits 2:0 – ADPS2:0: ADC Prescaler Select Bits

These bits determine the division factor between the system clock frequency and the input clock to the ADC.

Table 23-5. ADC Prescaler Selections

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

- Quando for utilizado um cristal de 20 MHz:
  - $20\text{MHz} / 2 = 10 \text{ MHz}$ ;
  - $20\text{MHz} / 4 = 5 \text{ MHz}$ ;
  - $20\text{MHz} / 8 = 2,5 \text{ MHz}$ ;
  - $20\text{MHz} / 16 = 1,25 \text{ MHz}$ ;
  - $20\text{MHz} / 32 = 625 \text{ kHz}$ ;
  - $20\text{MHz} / 64 = 312 \text{ kHz}$ ;
  - **$20\text{MHz} / 128 = 156 \text{ kHz}$** ;
- Recomendação (ATmega328) para 10 bits: entre 50 kHz e 200 kHz.

By default, the successive approximation circuitry requires an input clock frequency between 50 kHz and 200 kHz to get maximum resolution. If a lower resolution than 10 bits is needed, the input clock frequency to the ADC can be higher than 200 kHz to get a higher sample rate.

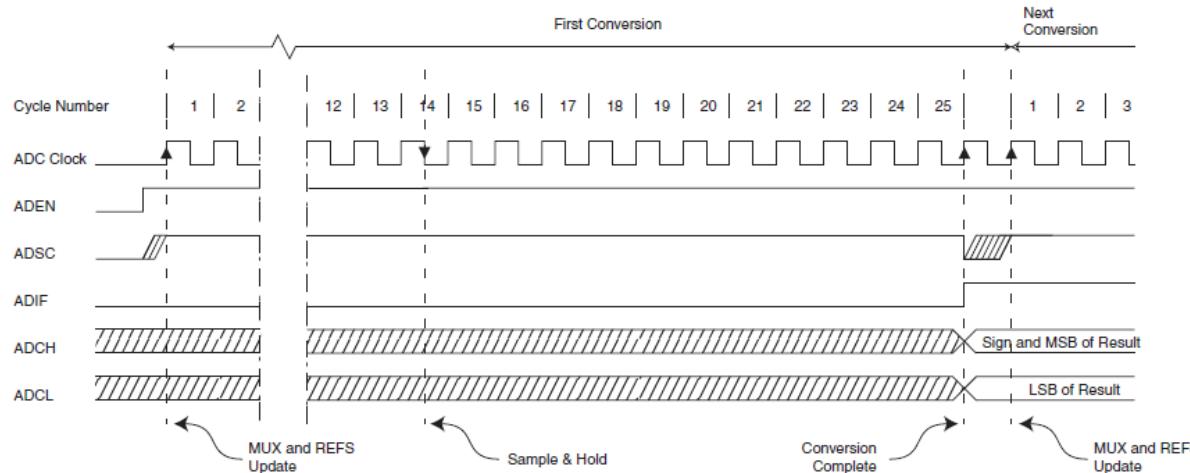
- A conversão simples do ADC ATMega328 funciona da seguinte forma:
  - Necessário inicialização a cada conversão;
  - Após finalizar a conversão, o bit **ADIF** do registrador **ADCSRA** é colocado em 1;
  - Iniciar uma conversão deve ligar o bit **ADSC** do registrador **ADCSRA**.

```
ADCSRA |= _BV(ADSC); //ADSC inicia a conversão;
while (!(ADCSRA & 0X10)); //ADIF == '0': avisa quando a conversão acabou.
```

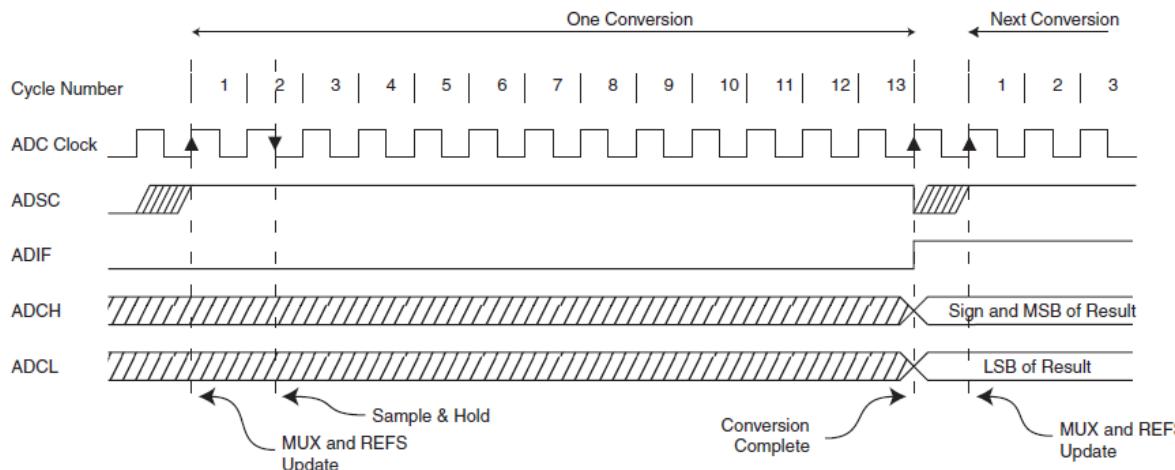
- Também existe o tipo de conversão contínua.

# ADC - ATMega328: Pulsos de Clock na Conversão

- A primeira conversão ocorre em 25 pulsos de clock.



- Da segunda conversão em diante são utilizados 13 pulsos de clock.



Informação  
influencia no  
tempo de  
conversão!

- O resultado da conversão é armazenado nos registradores **ADCL** e **ADCH**.

## 23.9.3 ADCL and ADCH – The ADC Data Register

### 23.9.3.1 ADLAR = 0

Bit	15	14	13	12	11	10	9	8	
(0x79)	-	-	-	-	-	-	ADC9	ADC8	ADCH
(0x78)	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	

### 23.9.3.2 ADLAR = 1

Bit	15	14	13	12	11	10	9	8	
(0x79)	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
(0x78)	ADC1	ADC0	-	-	-	-	-	-	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	

- ADLAR**: configura a forma de apresentação do resultado dos registradores ADCH e ADCL.

- O programador pode acessar a variável **ADC**, disponível pela biblioteca para ler o valor de ADCL e ADCH.

```
while(1){  
    ADCSRA |= _BV(ADSC);  
    while (!(ADCSRA & 0X10));  
  
    valor = ADC;  
}
```

- **Registrador ADMUX (ADC Multiplexer Selection)**

Bit	7	6	5	4	3	2	1	0	ADMUX
(0x7C)	REFS1	REFS0	ADLAR	-	MUX3	MUX2	MUX1	MUX0	
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **REFS0, REFS1:** Seleciona a tensão de referência.

Table 23-3. Voltage Reference Selections for ADC

REFS1	REFS0	Voltage Reference Selection
0	0	AREF, Internal $V_{ref}$ turned off
0	1	$AV_{CC}$ with external capacitor at AREF pin
1	0	Reserved
1	1	Internal 1.1V Voltage Reference with external capacitor at AREF pin

- **ADLAR:** Configura o ADCL e ADCH.

- **Registrador ADMUX (ADC Multiplexer Selection)**

Bit	7	6	5	4	3	2	1	0	
(0x7C)	REFS1	REFS0	ADLAR	-	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **MUX0, MUX1, MUX2, MUX3:** Seleciona o canal.

Table 23-4. Input Channel Selections

MUX3..0	Single Ended Input
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4
0101	ADC5
0110	ADC6
0111	ADC7
1000	ADC8 <sup>(1)</sup>

1001	(reserved)
1010	(reserved)
1011	(reserved)
1100	(reserved)
1101	(reserved)
1110	1.1V ( $V_{BG}$ )
1111	0V (GND)

- **Registrador ADCSRA (ADC Control and Status Register A)**

Bit	7	6	5	4	3	2	1	0	ADCSRA
(0x7A)	<b>ADEN</b>	<b>ADSC</b>	<b>ADATE</b>	<b>ADIF</b>	<b>ADIE</b>	<b>ADPS2</b>	<b>ADPS1</b>	<b>ADPS0</b>	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **ADEN:** Habilita o conversor.
- **ADSC:** Inicia a conversão.
  - Conversão simples: ‘1’ habilita cada conversão.
  - Conversão contínua: ‘1’ habilita todas as conversões.
- **ADATE:** Habilita Auto Trigger (disparo da conversão)
- **ADIF (ADC Interrupt Flag):** Indica quando a conversão finalizou.
- **ADIE:** Habilita a interrupção no final da conversão.
- **ADPS0, ADPS1, ADPS2:** Seleciona o fator de divisão do clock.

- Registrador ADCSRA (ADC Control and Status Register A)**

Bit	7	6	5	4	3	2	1	0	
(0x7B)	-	ACME	-	-	-	ADTS2	ADTS1	ADTS0	ADCSRB
Read/Write	R	R/W	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **ADTS0, ADTS1, ADTS2:** Seleciona a fonte de disparo.

Table 23-6. ADC Auto Trigger Source Selections

ADTS2	ADTS1	ADTS0	Trigger Source
0	0	0	Free Running mode
0	0	1	Analog Comparator
0	1	0	External Interrupt Request 0
0	1	1	Timer/Counter0 Compare Match A
1	0	0	Timer/Counter0 Overflow
1	0	1	Timer/Counter1 Compare Match B
1	1	0	Timer/Counter1 Overflow
1	1	1	Timer/Counter1 Capture Event

- **Exemplo:** Configurando o ADC.

```
ADMUX |= _BV(REFS0);
```

```
ADCSRA |= (1<<ADPS1) | _BV(ADEN);
```



```
ADMUX |= _BV(REFS0);
```

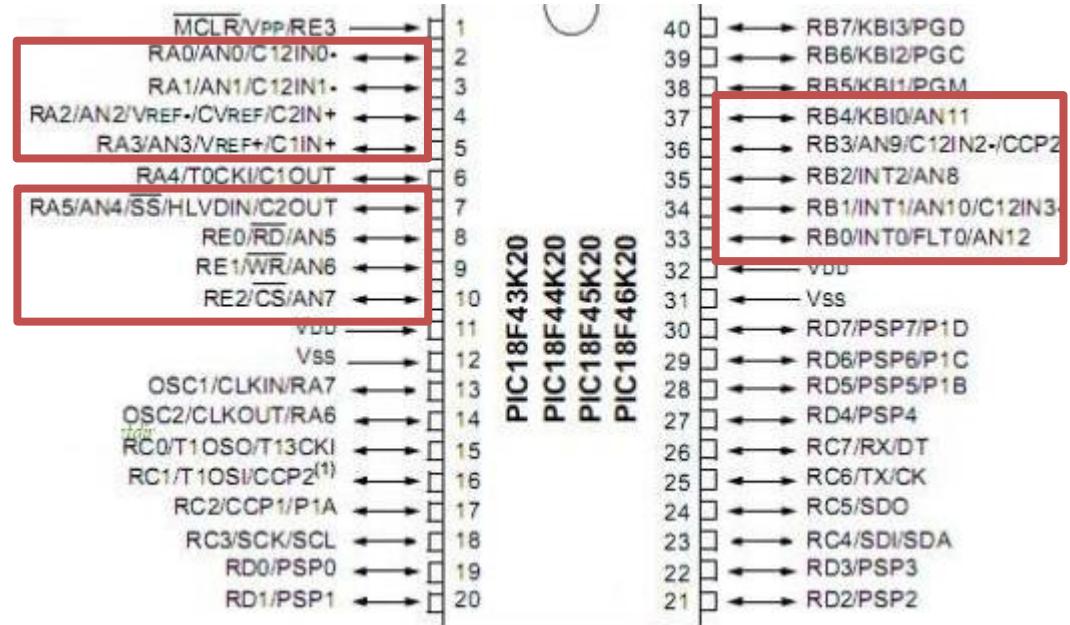
```
ADCSRA |= _BV(ADPS1) | _BV(ADEN);
```

```
ADMUX |= (1<<REFS0);
```

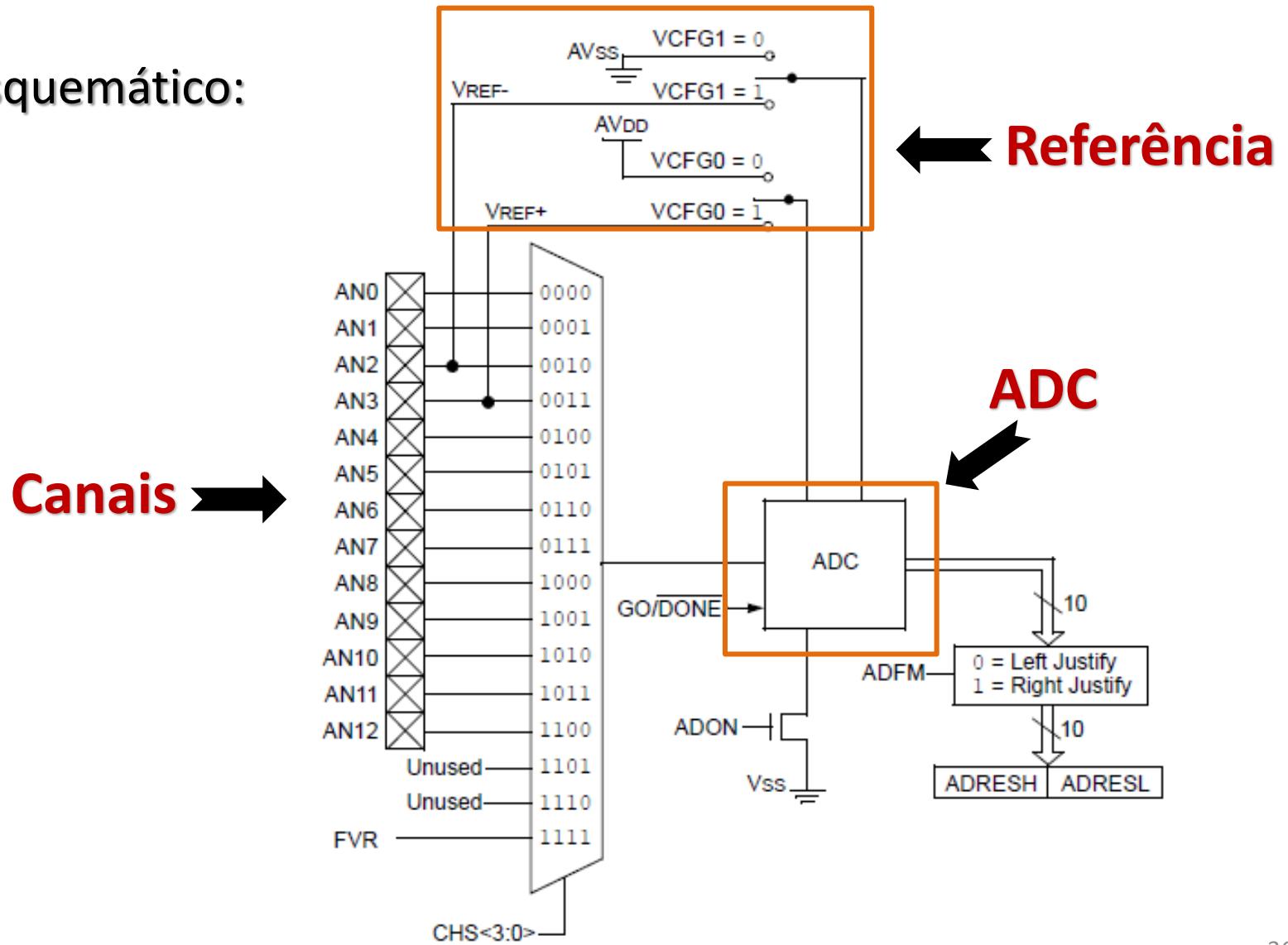
```
ADCSRA |= (1<<ADPS2) | (1<<ADPS1) | (1<<ADPS0) | (1<<ADEN);
```

- ❖ Diferentes soluções de configurar bits.

- Conversor A/D de aproximações sucessivas;
- Resolução de 10 bits;
- 13 canais (ANX);
- Faixa de tensão para conversão:
  - Referência VCC;
  - Referência externa.



- Esquemático:



- Registrador ADCON0 (A/D Control Register 0):**

REGISTER 19-1: ADCON0: A/D CONTROL REGISTER 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON
bit 7							bit 0

- **CHS<3:0>**: Selecionar o canal analógico;

0000 = AN0
0001 = AN1
0010 = AN2
0011 = AN3
0100 = AN4
0101 = AN5 <sup>(1)</sup>
0110 = AN6 <sup>(1)</sup>
0111 = AN7 <sup>(1)</sup>
1000 = AN8
1001 = AN9
1010 = AN10
1011 = AN11
1100 = AN12
1101 = Reserved
1110 = Reserved
1111 = FVR (1.2 Volt Fixed Voltage Reference) <sup>(2)</sup>

- **Registrador ADCON0 (A/D Control Register 0):**

REGISTER 19-1: ADCON0: A/D CONTROL REGISTER 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON
bit 7							bit 0

- **GO/DONE:** Indica o status do processo de conversão;
  - ‘1’: Conversão em andamento;
  - ‘0’: Conversão concluída.
- **ADON:** Habilita o conversor AD.
  - ‘1’: Conversor habilitado;
  - ‘0’: Conversor desabilitado.

- **Registrador ADCON1 (A/D Control Register 1):**

REGISTER 19-2: ADCON1: A/D CONTROL REGISTER 1

U-0	U-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	—	VCFG1	VCFG0	—	—	—	—
bit 7							bit 0

- **VCFG1, VFD0:** Seleciona a tensão de referência.

- VCFG1 == '1':  $V_- := V_{ref-}$
- VCFG1 == '0':  $V_- = V_{ss}$ ;
- VCFG0 == '1':  $V_+ := V_{ref+}$
- VCFG0 == '0':  $V_+ = V_{dd}$ .

- **Registrador ADCON2 (A/D Control Register 2):**

**REGISTER 19-3: ADCON2: A/D CONTROL REGISTER 2**

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7							bit 0

- **ADFM:** Seleciona o formato do resultado (m328: ADLAR).
- **ACTQ:** Seleciona o tempo de aquisição do dados.
  - Nos µC PIC, existe um capacitor ligado ao canal ADC selecionado durante o tempo de amostragem para evitar que variações na tensão aplicada ao canal influencie na amostragem.
  - Para que seja realizada a amostragem, consequentemente, a conversão correta, é necessário que o ADC espere o capacitor carregar totalmente antes de iniciar uma nova conversão.
  - $T_{DA}$ : tempo de adequação do carregador.

- Registrador ADCON2: **ACTQ**.

000 = 0<sup>(1)</sup>

001 = 2 TAD

010 = 4 TAD

011 = 6 TAD

100 = 8 TAD

101 = 12 TAD

110 = 16 TAD

111 = 20 TAD

- ❖ Para um conversor de 10 bits recomenda-se escolher um tempo ACTQ de  $16 \cdot T_{DA}$  ou, no mínimo, de  $12 \cdot T_{DA}$  (10bits + 2 pulsos).

- Registrador ADCON2 (A/D Control Register 2):

**REGISTER 19-3: ADCON2: A/D CONTROL REGISTER 2**

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7				bit 0			

- **ADCS**: Seleciona o fator de divisão do clock.

000 = Fosc/2
001 = Fosc/8
010 = Fosc/32
011 = FRC <sup>(1)</sup>
100 = Fosc/4
101 = Fosc/16
110 = Fosc/64
111 = FRC <sup>(1)</sup>

- Registradores de armazenamento do resultado:**

**REGISTER 19-4: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 0**

| R/W-x  |
|--------|--------|--------|--------|--------|--------|--------|--------|
| ADRES9 | ADRES8 | ADRES7 | ADRES6 | ADRES5 | ADRES4 | ADRES3 | ADRES2 |
| bit 7  |        |        |        |        |        |        | bit 0  |

**REGISTER 19-5: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 0**

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
ADRES1	ADRES0	—	—	—	—	—	—
bit 7							bit 0

**REGISTER 19-6: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 1**

R/W-x	R/W-x						
—	—	—	—	—	—	ADRES9	ADRES8
bit 7							bit 0

**REGISTER 19-7: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 1**

| R/W-x  |
|--------|--------|--------|--------|--------|--------|--------|--------|
| ADRES7 | ADRES6 | ADRES5 | ADRES4 | ADRES3 | ADRES2 | ADRES1 | ADRES0 |
| bit 7  |        |        |        |        |        |        | bit 0  |

- **ANSEL (Analog Select Register)**: Habilita e desabilita entrada digital.

## REGISTER 10-2: ANSEL: ANALOG SELECT REGISTER 1

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ANS7 <sup>(1)</sup>	ANS6 <sup>(1)</sup>	ANS5 <sup>(1)</sup>	ANS4	ANS3	ANS2	ANS1	ANS0
bit 7						bit 0	

bit 7

### **ANS7: RE2 Analog Select Control bit<sup>(1)</sup>**

1 = Digital input buffer of RE2 is disabled  
0 = Digital input buffer of RE2 is enabled

bit 6

### **ANS6: RE1 Analog Select Control bit<sup>(1)</sup>**

**1** = Digital input buffer of RE1 is disabled  
**0** = Digital input buffer of RE1 is enabled

bit 5

### **ANS5: RE0 Analog Select Control bit<sup>(1)</sup>**

1 = Digital input buffer of RE0 is disabled  
0 = Digital input buffer of RE0 is enabled

bit 4

#### **ANS4: RA5 Analog Select Control bit**

1 = Digital input buffer of RA5 is disabled  
0 = Digital input buffer of RA5 is enabled

bit 3

### **ANS3: RA3 Analog Select Control bit**

1 = Digital input buffer of RA3 is disabled  
0 = Digital input buffer of RA3 is enabled

## **ANS2: RA2 Analog Select Control bit**

**1** = Digital input buffer of RA2 is disabled  
**0** = Digital input buffer of RA2 is enabled

## **ANS1: RA1 Analog Select Control bit**

1 = Digital input buffer of RA1 is disabled  
0 = Digital input buffer of RA1 is enabled

#### **ANS0: RA0 Analog Select Control bit**

**1** = Digital input buffer of RA0 is disabled  
**0** = Digital input buffer of RA0 is enabled

- ANSELH (Analog Select Register):** Habilita e desabilita entrada digital.

REGISTER 10-3: ANSELH: ANALOG SELECT REGISTER 2

U-0	U-0	U-0	R/W-1 <sup>(1)</sup>				
—	—	—	ANS12	ANS11	ANS10	ANS9	ANS8
bit 7							bit 0

**Legend:**

R = Readable bit

-n = Value at POR

W = Writable bit

'1' = Bit is set

U = Unimplemented bit, read as '0'

'0' = Bit is cleared

x = Bit is unknown

- |         |  |
|---------|--|
| bit 7-5 | <b>Unimplemented:</b> Read as '0'  |
| bit 4   | <b>ANS12:</b> RB0 Analog Select Control bit<br>1 = Digital input buffer of RB0 is disabled<br>0 = Digital input buffer of RB0 is enabled |
| bit 3   | <b>ANS11:</b> RB4 Analog Select Control bit<br>1 = Digital input buffer of RB4 is disabled<br>0 = Digital input buffer of RB4 is enabled |
| bit 2   | <b>ANS10:</b> RB1 Analog Select Control bit<br>1 = Digital input buffer of RB1 is disabled<br>0 = Digital input buffer of RB1 is enabled |
| bit 1   | <b>ANS9:</b> RB3 Analog Select Control bit<br>1 = Digital input buffer of RB3 is disabled<br>0 = Digital input buffer of RB3 is enabled  |
| bit 0   | <b>ANS8:</b> RB2 Analog Select Control bit<br>1 = Digital input buffer of RB2 is disabled<br>0 = Digital input buffer of RB2 is enabled  |

1. Configurar pino/porta:
  - Habilitar a porta como entrada (**TRIS**);
  - Desabilitar o pino digital (**ANSEL**);
2. Configurar o conversor:
  - a. Selecionar o fator de conversão do clock (**ADCS - ADCON2**);
  - b. Configurar a tensão de referência (**VCFG - ADCON1**);
  - c. Selecionar o canal ADC (**CHS - ADCON0**);
  - d. Habilitar o conversor (**ADON – ADCON0**);
  - e. Selecionar o formato do resultado (**ADFM – ADCON2**);
3. Configurar a interrupção do ADC (opcional);
4. Esperar o tempo de aquisição do dado;
5. Iniciar a conversão (**GO/DONE – ADCON0**).
6. Esperar o conversor converter o dado;
7. Ler o resultado do ADC (**ADRESH, ADRESL**);
8. Limpar o flag de interrupção.

## 1. Configurar pino/porta:

- Habilitar a porta como entrada (**TRIS**);
- Desabilitar o pino digital (**ANSEL**);

```
#define _XTAL_FREQ 16000000
#include <xc.h>
#pragma config FOSC = INTIO67

unsigned int valor = 0;

void main(void) {

    TRISA |= 0b00000001;
    ANSEL |= 0b00000001;
    ANSELH = 0;
```

```
#define _XTAL_FREQ 16000000
#include <xc.h>
#pragma config FOSC = INTIO67

unsigned int valor = 0;

void main(void) {

    TRISA |= 0b00000001;
    ANSELbits.ANS0 = 1;
    ANSELH = 0;
```

## 2. Configurar o conversor:

- a. Selecionar o fator de conversão do clock (**ADCS - ADCON2**);
- b. Configurar a tensão de referência (**VCFG - ADCON1**);
- c. Selecionar o canal ADC (**CHS - ADCON0**);
- d. Habilitar o conversor (**ADON – ADCON0**);

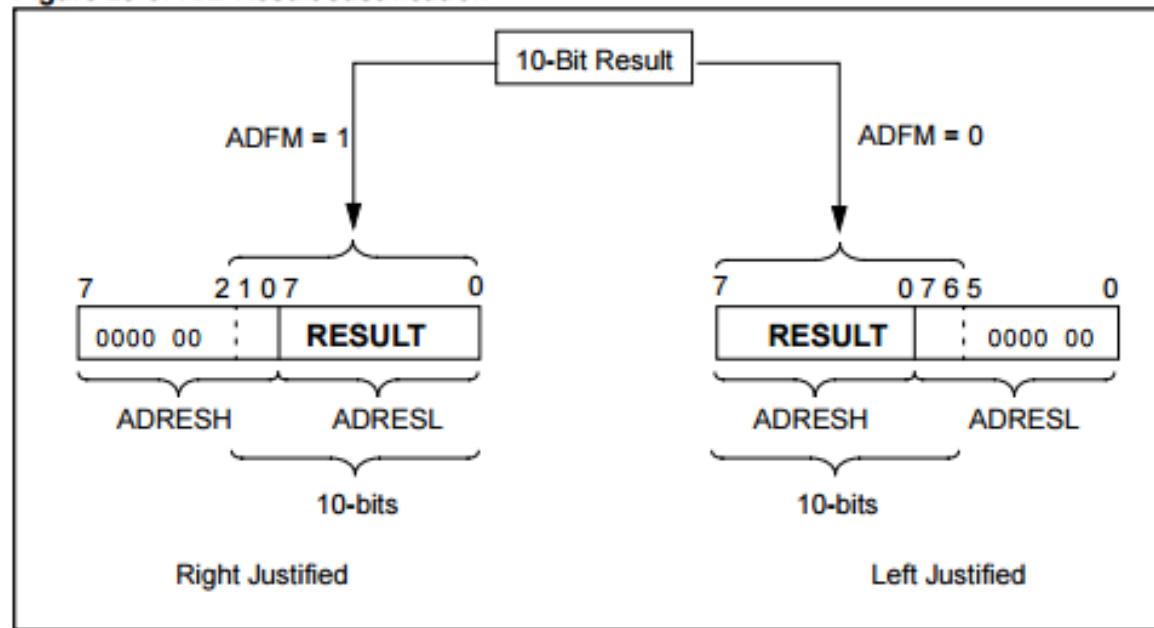
```
//ADCS = 101 - Clock Fosc/16
ADCON2 |= 0b00000101;
ADCON2 &= 0b11111101;
//VCFG0 = 0; VCFG1 = 0; Tensao Vdd
ADCON1 &= 0b11001111;
//CHS = 0000 - Canal 0: AN0
ADCON0 &= 0b11000011;
//Habilitar o ADC
ADCON0 |= 0b00000001;
```

## 2. Configurar o conversor:

- e. Selecionar o formato do resultado (**ADFM – ADCON2**);

```
//ADFM = 1 (direita) - Formato do resultado  
ADCON2 |= 0b10000000;
```

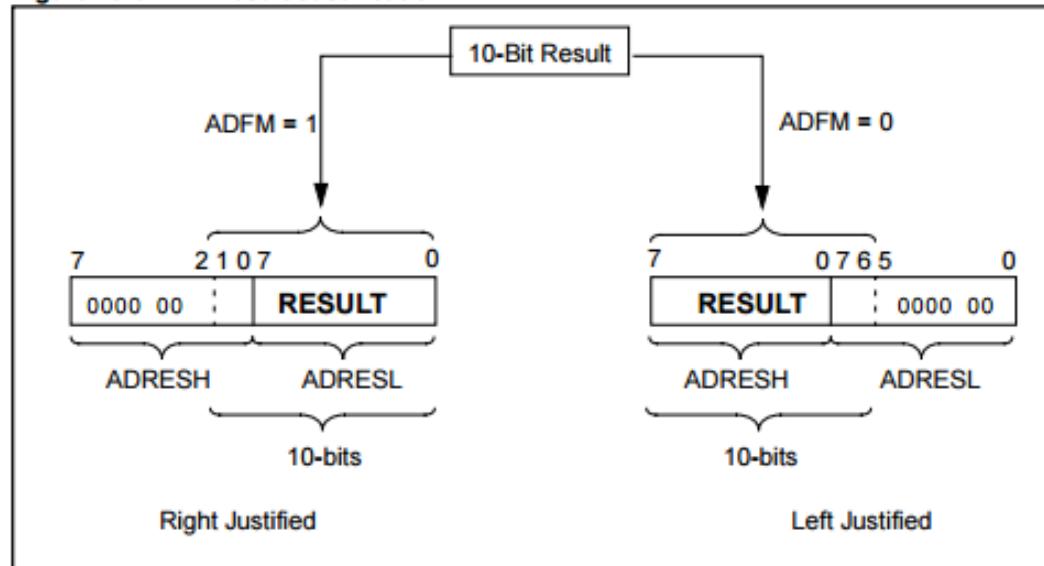
Figure 23-6: A/D Result Justification



- Situação1:** Dado que o conversor é de 10 bits e o ADFM=1, qual a faixa de valor de “result”?

```
//Resultado da conversão
result = ((unsigned int)(ADRESH << 8) | ADRESL);
```

Figure 23-6: A/D Result Justification



- **Situação1:** Dado que o conversor é de 10 bits e o ADFM=1, qual a faixa de valor de “result”?

```
//Resultado da conversão  
result = ((unsigned int)(ADRESH << 8) | ADRESL);
```

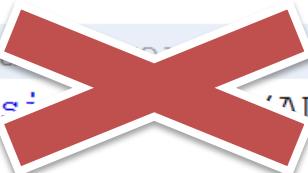
➤ **result** varia entre **0 e 1023**.

- **Situação2:** Se ADFM=0, qual a faixa de valor de “result”?

```
//Resultado da conversão  
result = ((unsigned int)(ADRESH << 8) | ADRESL);
```

- **Situação1:** Dado que o conversor é de 10 bits e o ADFM=1, qual a faixa de valor de “result”?
  - result varia entre **0 e 1023**.
- **Situação2:** Se ADFM=0, qual a faixa de valor de “result”?
  - O valor de result está sendo calculado errado!

```
//Resultado da conversão  
result = ((unsigned int)(ADRESH << 8) | ADRESL);
```



4. Esperar o tempo de aquisição do dado;
5. Iniciar a conversão (**GO/DONE – ADCON0**).
6. Esperar o conversor converter o dado;
7. Ler o resultado do ADC (**ADRESH, ADRESL**);

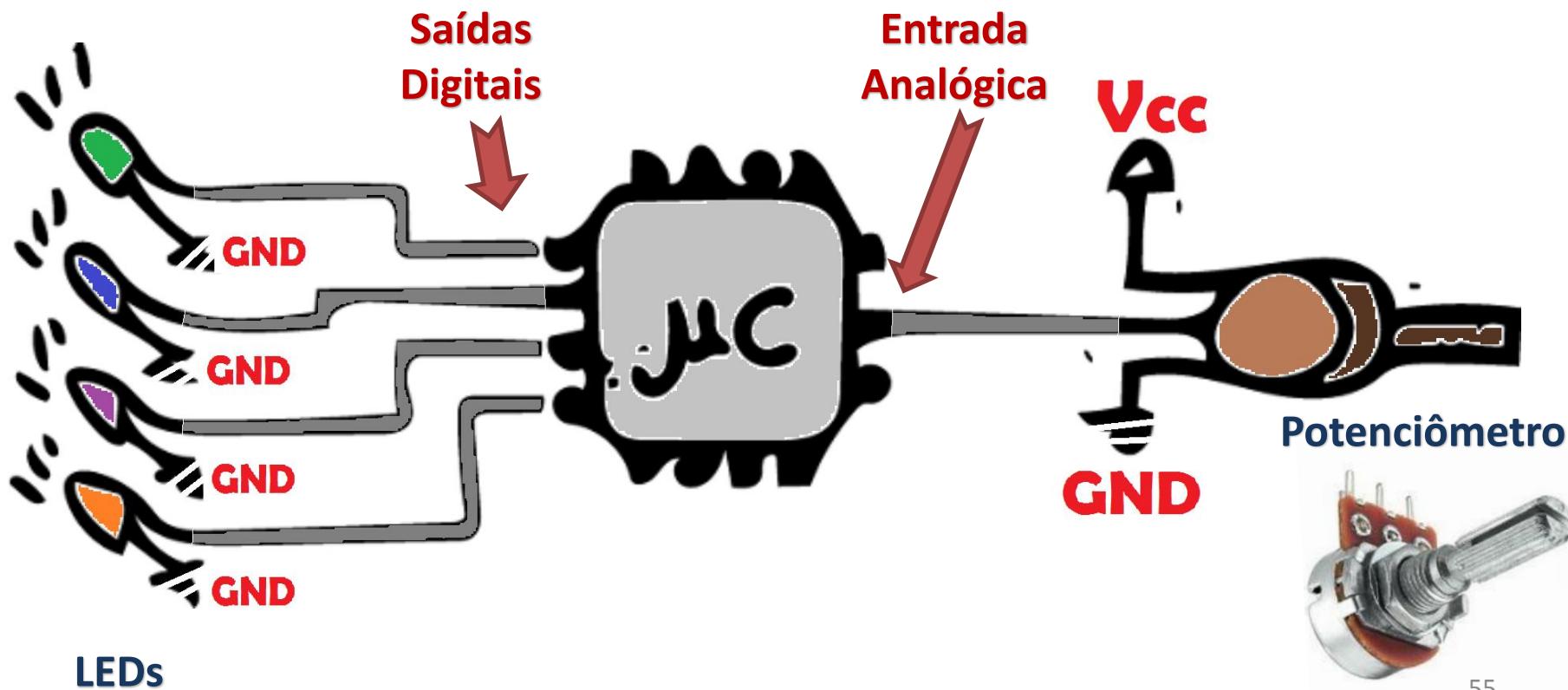
```
while(1) {
    __delay_ms(1);
    //Iniciar a conversão;
    ADCON0bits.GO_nDONE = 1;
    //Esperar a conversão terminar;
    while(ADCON0bits.GO_nDONE == 1);

    //Resultado da conversão
    valor = ((unsigned int)(ADRESH << 8) | ADRESL);

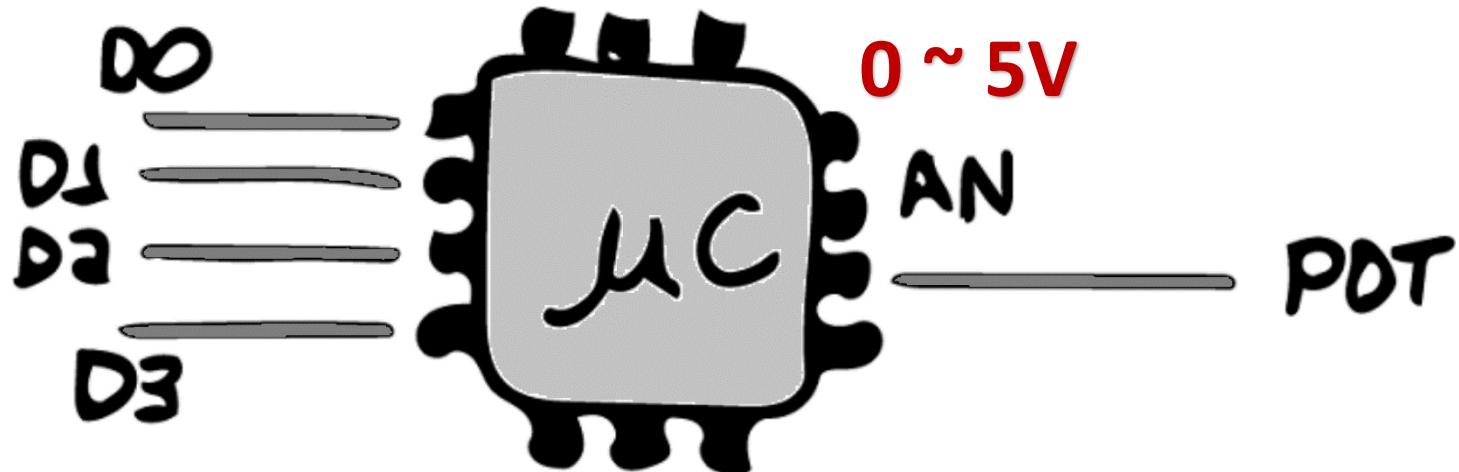
    //valor lido na faixa de 0 a 1023
}
```

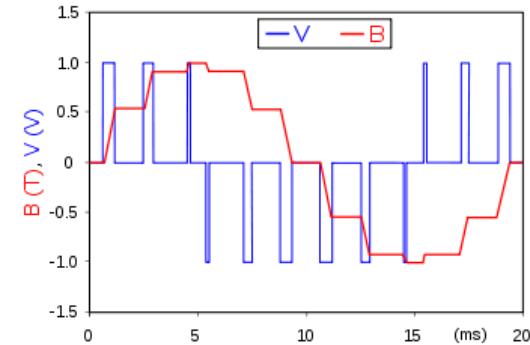
# Exercícios

- Programe o microcontrolador para receber um valor analógico proveniente de um potenciômetro e acenda a barra de LEDs (VU de LEDs) de acordo com o valor recebido.

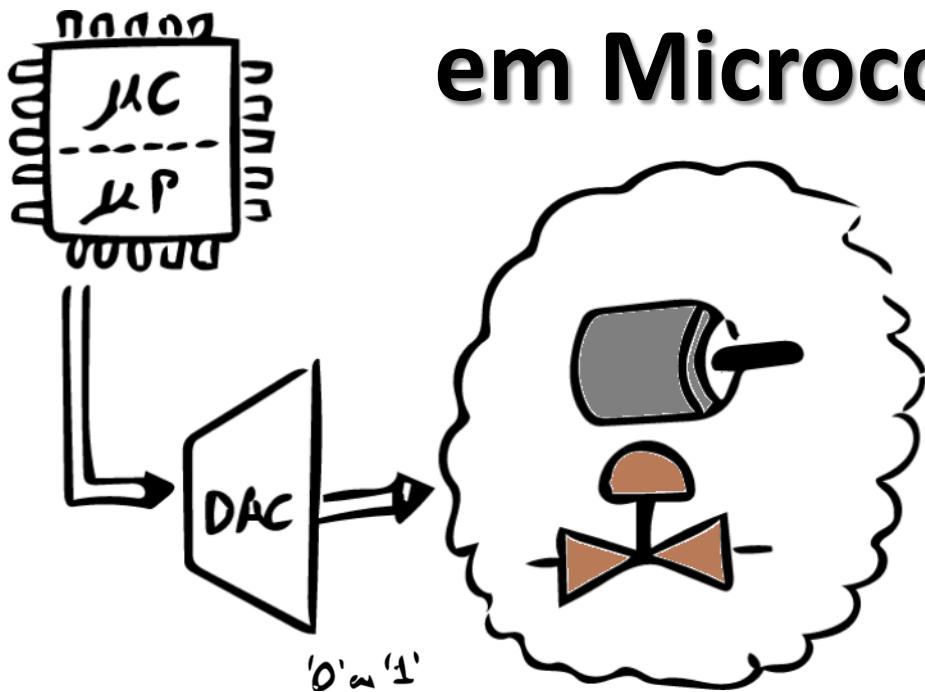


- O valor recebido será de 0 a 5V, a barra de LEDs segue a lógica:
  - 0 a ~1,25: Acender LED0;
  - ~1,26 a ~2,5: Acender LED0, LED1;
  - ~2,51 a ~3,75: Acender LED0, LED1, LED2;
  - ~3,76 a 5: Acender LED0, LED1, LED2, LED3;

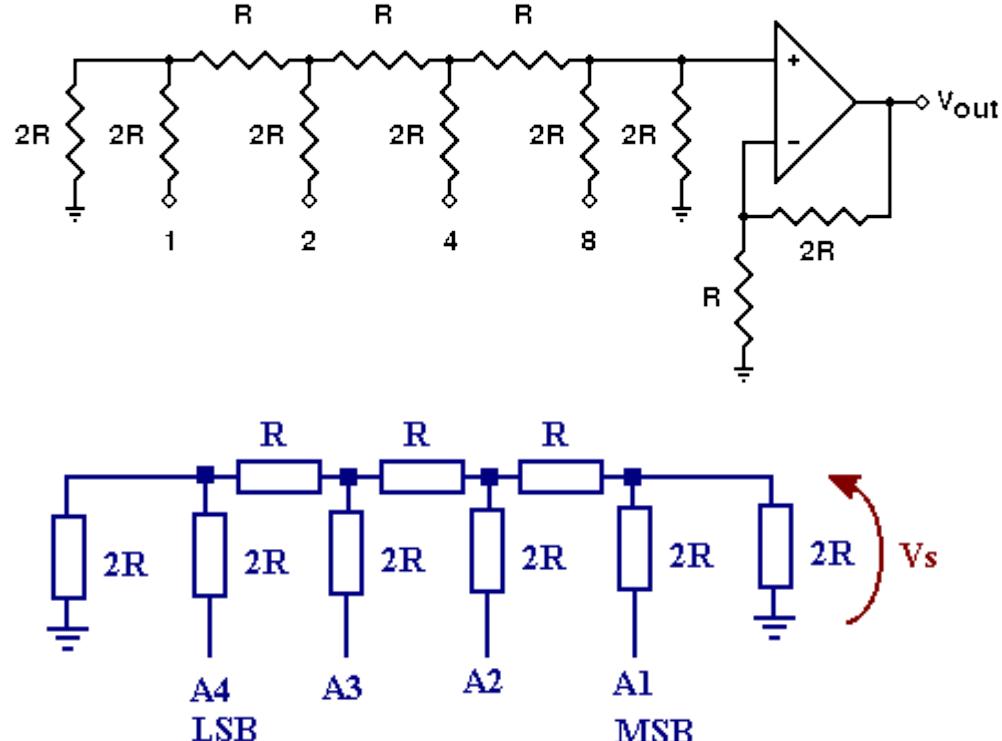




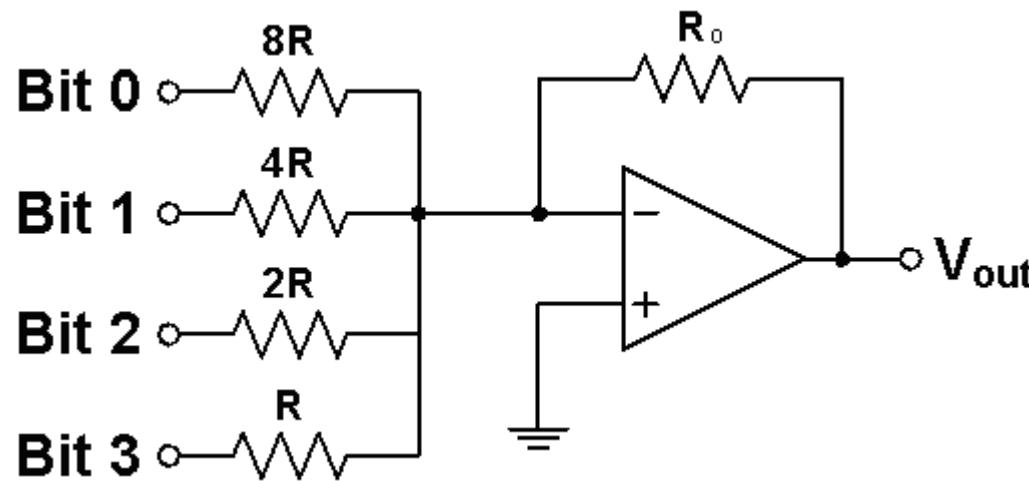
# Digital → Analógico em Microcontroladores



- Existem vários circuitos que recebem um sinal digital, 0 V ('0') ou 5 V ('1'), e convertem em um sinal analógico, 0V à 10V.
- Exemplos:
  - Conversor R-2R



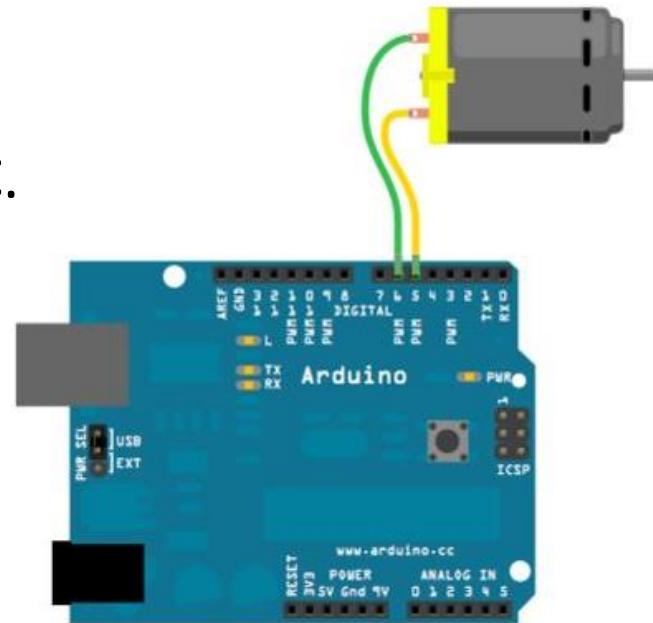
- Exemplos:
  - Conversor Resistências Ponderadas



- Existem microcontroladores que contêm conversores DA (digital → analógico). Porém, a maioria não contém.

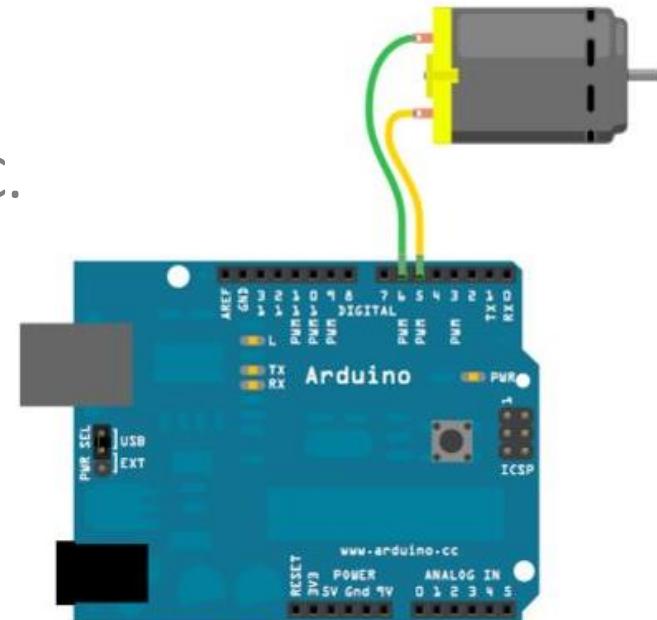
- **Problema:** Desenvolver um sistema microcontrolado pelo ATMega328 para controlar a velocidade de rotação de um motor DC?

Obs.: ATMega328 não tem DAC.



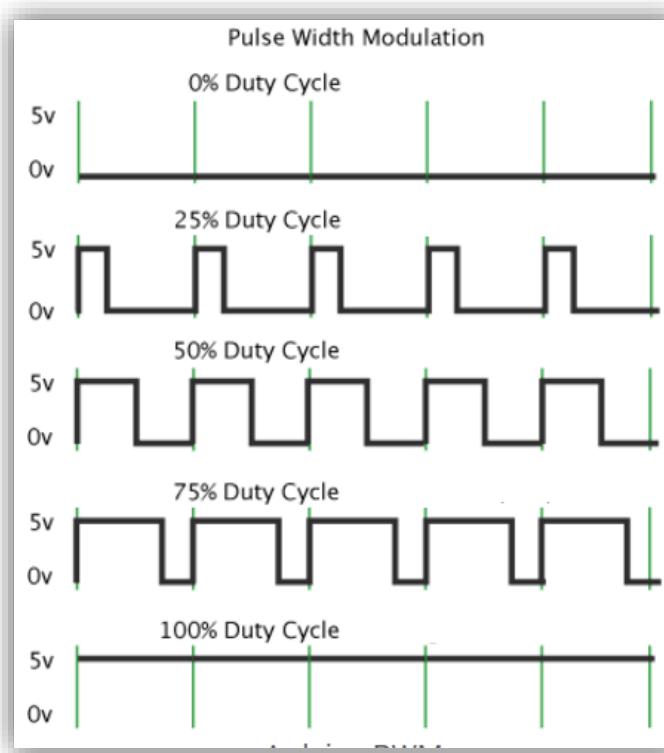
- **Problema:** Desenvolver um sistema microcontrolado pelo ATMega328 para controlar a velocidade de rotação de um motor DC?

Obs.: ATMega328 não tem DAC.



- **Solução:** Utilizar a técnica PWM.

- PWM é uma técnica utilizada para variar alterar uma onda quadrada com o intuito de variar o valor médio da onda.
- Manter a frequência da onda quadrada fixa e alterar o tempo (**duty cycle**) que o sinal da onda fica em nível lógico alto.



- Dado que o valor digital ('1') aplicado é de 5V (Vcc), o valor médio da onda é dado por:

$$V_{out} = \left( \frac{\text{duty cycle}}{100} \right) * V_{cc}$$
$$V_{out} = \left( \frac{\text{duty cycle}}{100} \right) * 5$$

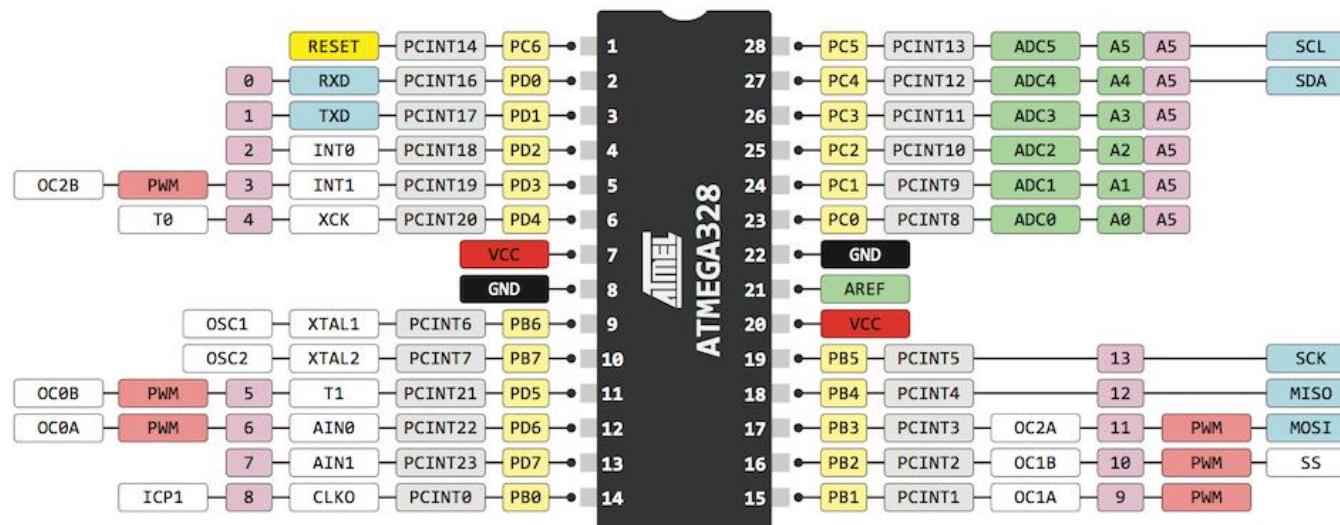
- **Duty cycle:** valor do ciclo ativo do PWM em %;
- **Vout:** valor médio;

- É possível variar a potência de um motor sem variar a tensão aplicada ao motor;
- A velocidade e potência do motor depende da tensão média aplicada ao motor;
- O controle da velocidade do motor através de um sinal PWM é comumente utilizado;

# ATMega328: Pinos de PWM

THE  
DEFINITIVE  
**ATMEGA328**  
**&Arduino**  
PINOUT DIAGRAM

■	GND
■	Power
■	Control
■	Physical Pin
■	Port Pin
■	Pin Function
■	Digital Pin
■	Analog Related Pin
■	PWM Pin
■	Serial Pin
■	IDE

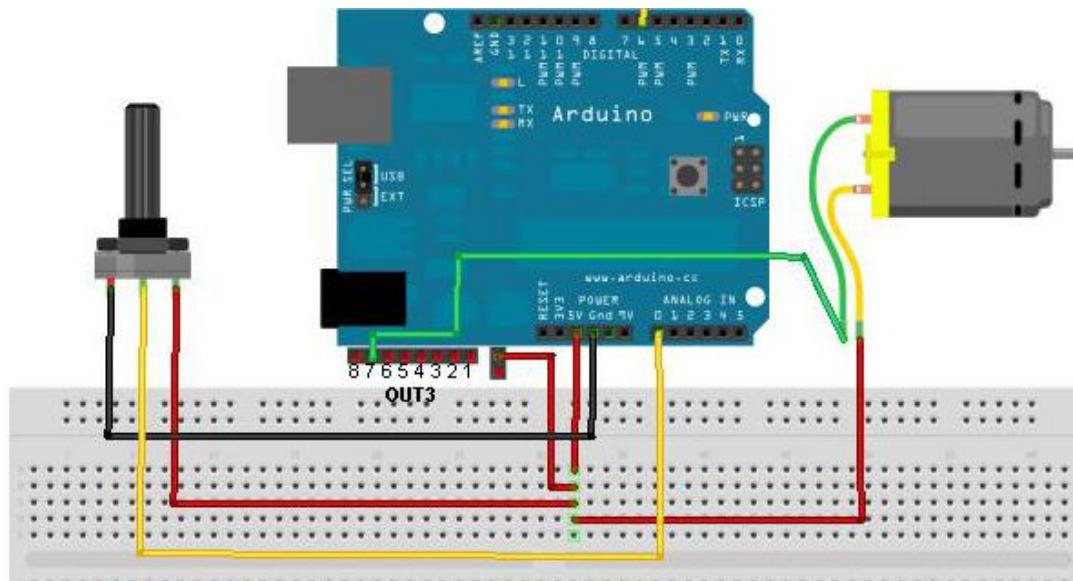


- O sinais de PWM são gerados por temporizadores/contadores dos µC.

# **Exercícios: Iteração com o mundo analógico**

# Exercício

- Projete um sistema µcontrolado, baseado no µC ATMega328, para controlar a velocidade de um motor DC a partir do valor de um potenciômetro. Quando maior a tensão no potenciômetro, maior a velocidade do motor.



# Referências

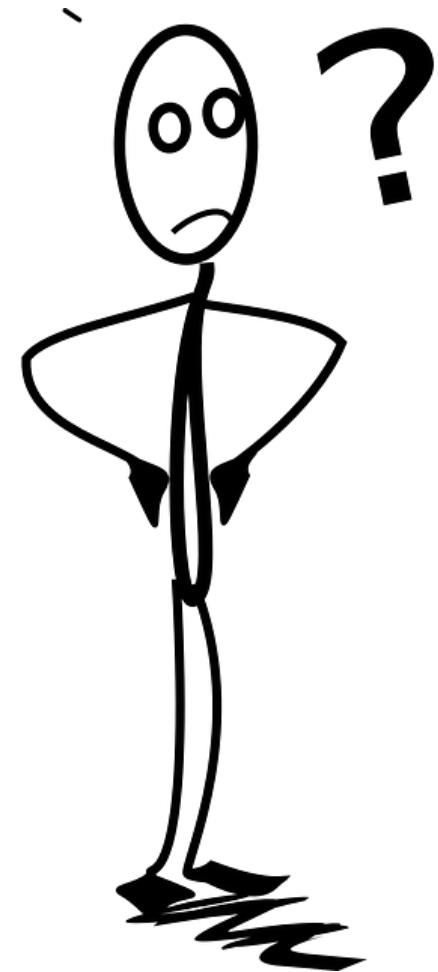
- Embarcados. **Trazendo o mundo real para dentro do processador - Conversor A/D.** Disponível em:  
<http://www.embarcados.com.br/conversor-a-d/>.
- Embarcados. **Arduino UNO - Taxa de amostragem do conversor A/D.** Disponível em: <http://www.embarcados.com.br/arduino-taxa-de-amostragem-conversor-ad/>.
- Newton Braga. **Implementando um ADC com um Microcontrolador, AO e resistores (MIC085).** Disponível em:  
<http://www.newtoncbraga.com.br/index.php/microcontroladores/143-tecnologia/6375-mic085>.
- Newton Braga. **Controle PWM de Motor DC (MEC139).** Disponível em: <http://www.newtoncbraga.com.br/index.php/robotica/5534-mec139>.

# Referências

- **Datasheet:** ATMega328, Atmel.
- **Datasheet:** PIC18F45K20, Microchip.

- <http://www.embarcados.com.br/condicionamento-de-sinais-analogicos/>
- <http://www.embarcados.com.br/condicionamento-de-sinais-analogicos-parte-2/>
- <http://www.embarcados.com.br/condicionamento-de-sinais-analogicos-parte-3/>

# Dúvidas?



Dúvidas por email:

[heitorm@dca.ufrn.br](mailto:heitorm@dca.ufrn.br)