

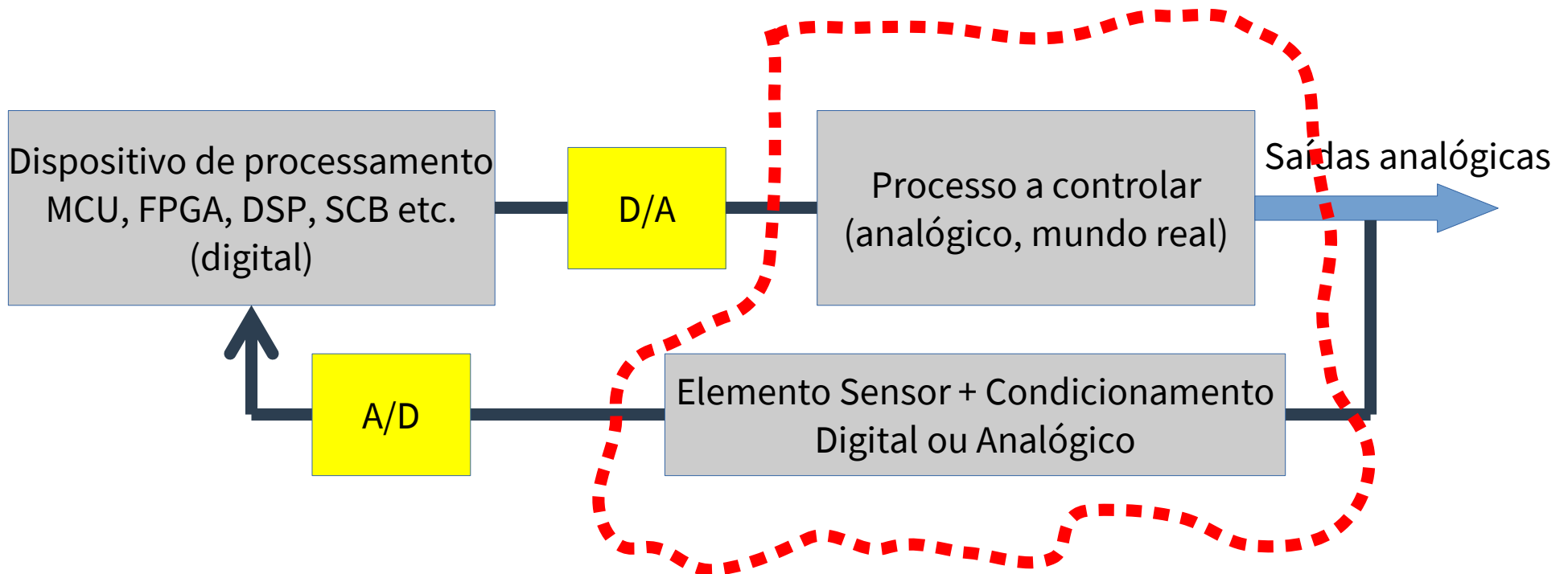
SISTEMAS EMBARCADOS

PROF. JOSENALDE OLIVEIRA

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - UFRN



Aquisição de Dados em Sistemas Embarcados

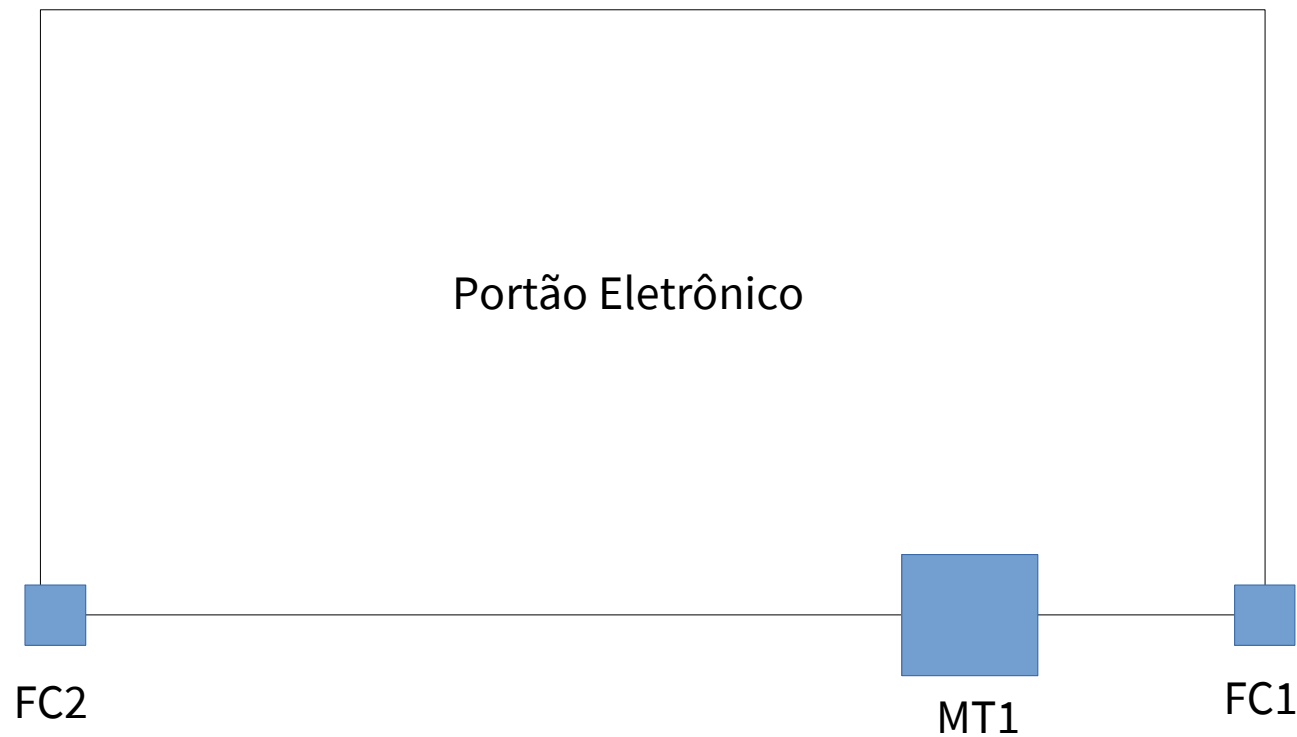


Conversor Analógico-Digital: representar grandeza analógica por número associado para manipulação digital

Conversão Digital-Analógico: PWM é uma possibilidade, mas existem placas com tensão DC equivalente em faixas: -10V :10V

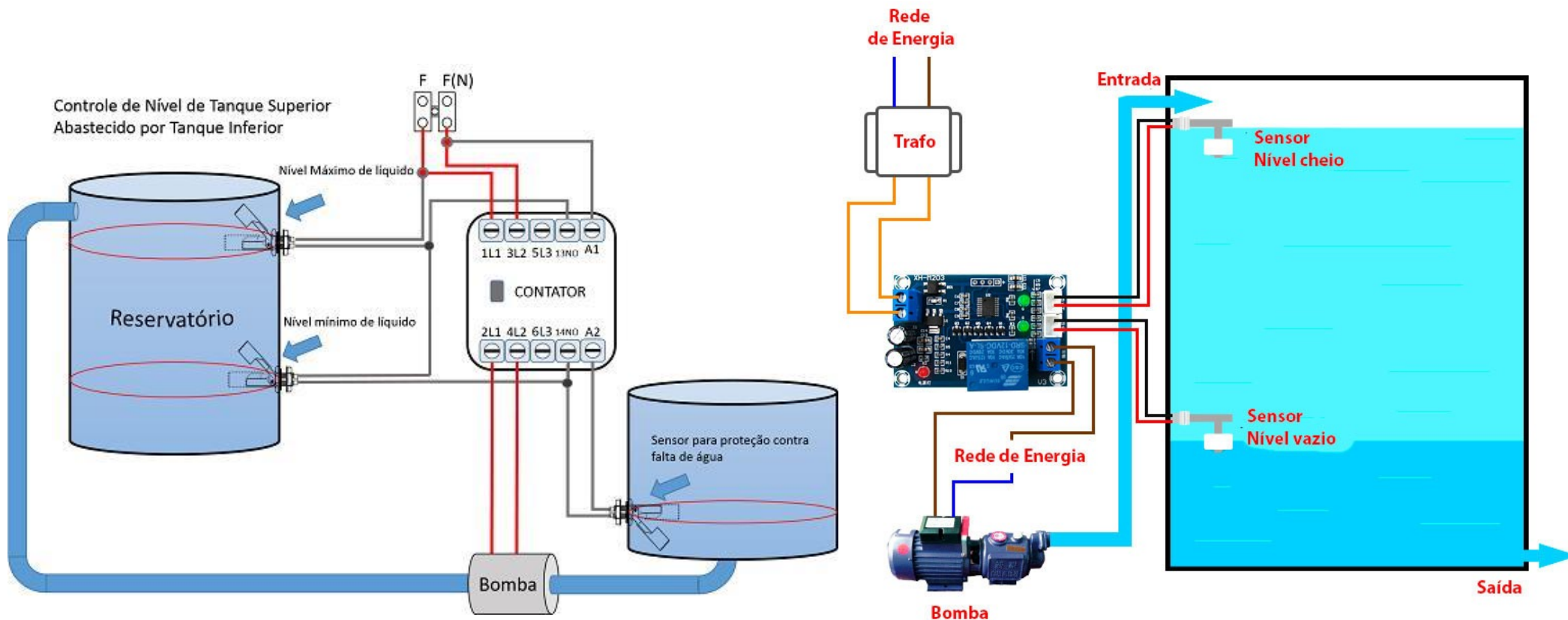
Aquisição de Dados em Sistemas Embarcados

Exemplo : Entradas e Saídas Digitais



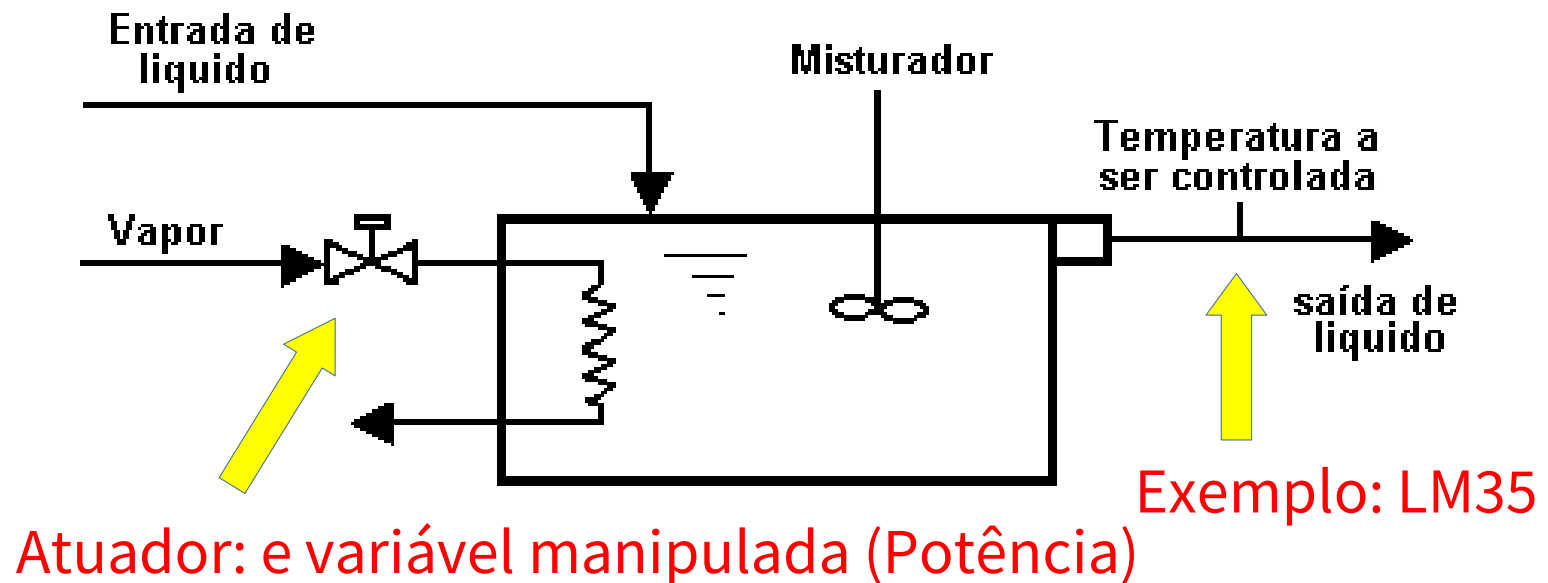
Aquisição de Dados em Sistemas Embarcados

Exemplo : Entradas e Saídas Digitais: Controle de Nível



Aquisição de Dados em Sistemas Embarcados

**Exemplo : E se além de ligar/desligar, controlar nível, desejar também
Controlar temperatura do líquido no tanque?**



**Normalmente sensores e atuadores com variáveis analógicas: válvula
4 a 20 mA, sensor de nível e temperatura (0 – 10 V), resistência (0 - 10V)**

http://www.ebah.com.br/content/ABAAAA_TUAI/teoria-controle

Aquisição de Dados em Sistemas Embarcados

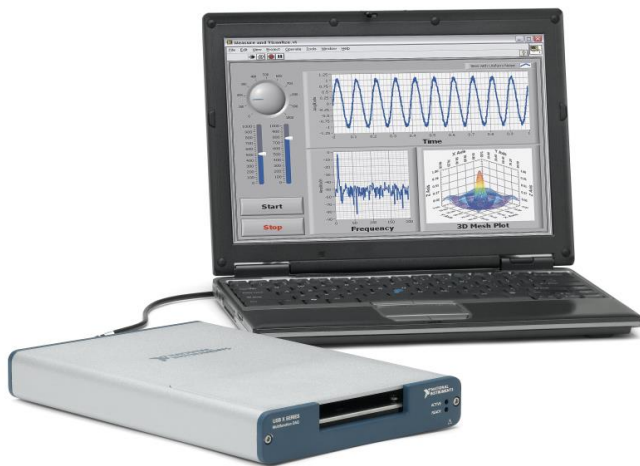
Variáveis analógicas comuns: nível, temperatura, pressão, posição, luminosidade, velocidade, pH, EC, distância, massa, ...

Para cada variável, tipos de sensores diferentes (magnéticos, capacitivos, Indutivos, resistivos...), com faixas de leituras diferentes, precisão, tipo de saída, Faixa de alimentação, linearidade, frequência de operação tempo de resposta

TRANSDUTOR: dispositivo que trabalha junto com o sensor e converte/transforma/adequa o valor lido para compatibilidade com a unidade de Processamento e ou meio de transmissão: ex: $1\text{mV}/^{\circ}\text{C}$

Aquisição de Dados em Sistemas Embarcados

**Comercialmente existem placas de aquisição e envio de dados (AD-DA):
National Instruments, Advantech etc.**

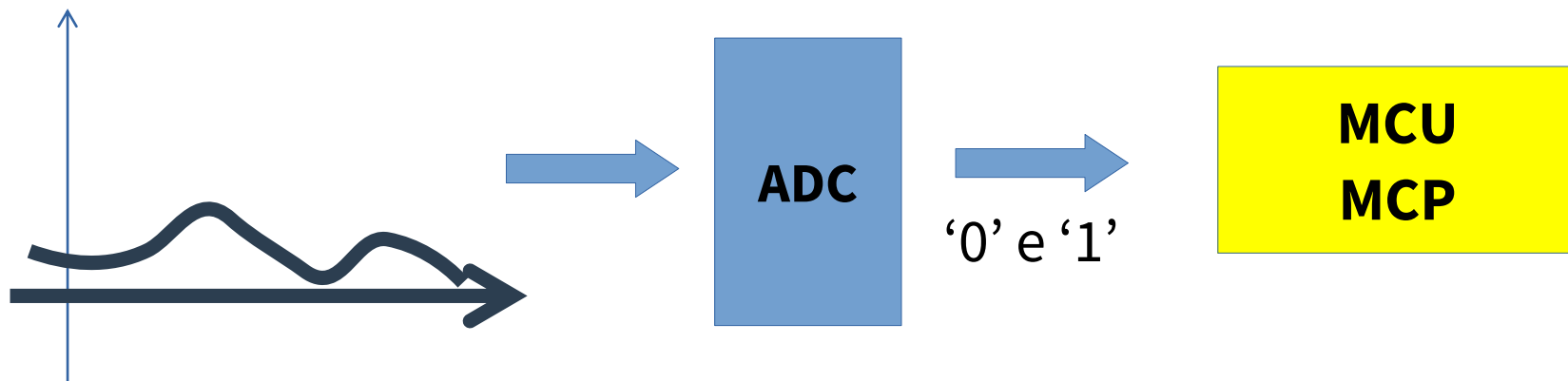


1.638,00 Euros

**16 Canais Analógicos de Entrada, 2MSamples/s = 2MHz, 16 bits resolução, 24 entradas digitais bi-dir
02 canais analógicos de saída (-10, 10V), 04 temporizadores/contadores**

Soluções low-cost para AD e processo AD

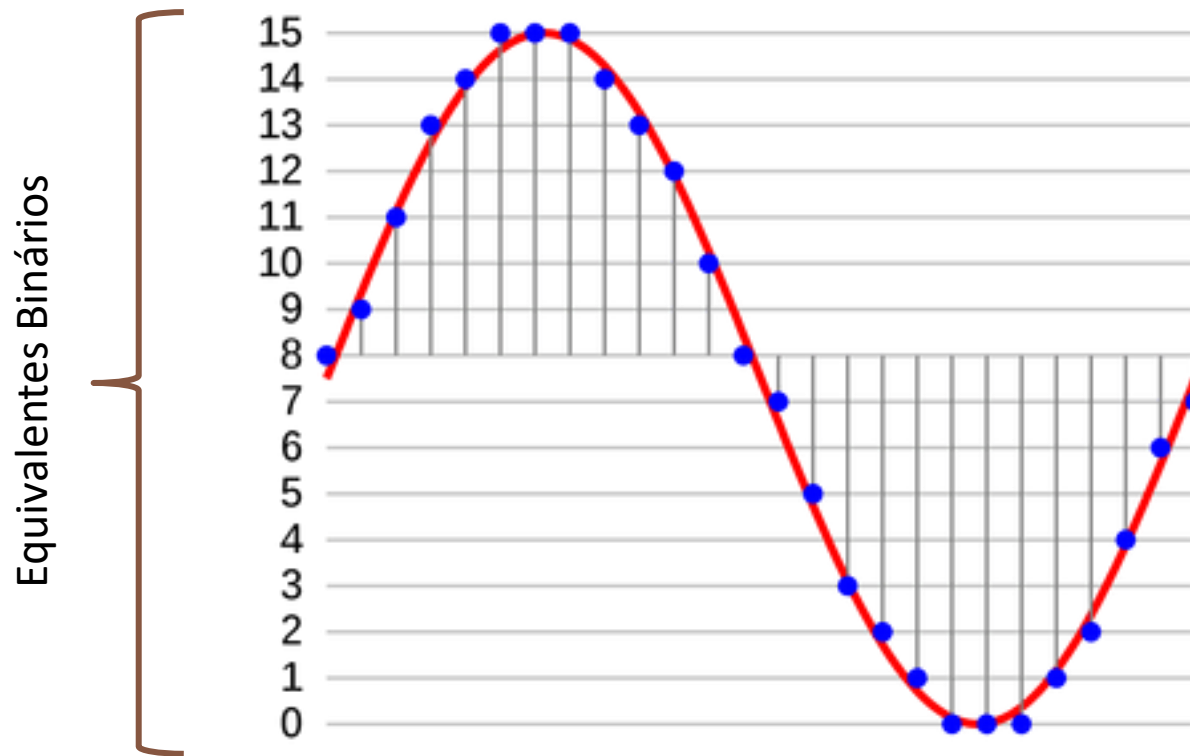
- ATmega328 (06 canais no Arduino UNO: A0-A5): AD de 10 bits de resolução
- ATmega2560 (16 canais no Arduino Mega), mesma resolução



Existem várias técnicas de ADC: rampa, **aproximações sucessivas**, flash, sigma-delta etc.

Soluções low-cost para AD e processo AD

- ADC converte sinal analógico no tempo ($x(t)$) em sinal amostrado no tempo $x(kT)$, onde T é o tempo de amostragem. Estes valores são quantizados em conjuntos de valores inteiros e depois codificados em valores digitais.



Soluções low-cost para AD e processo AD

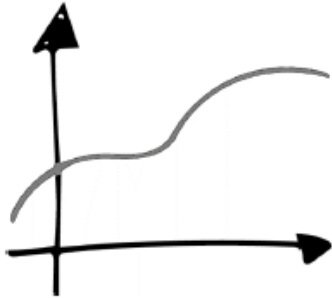
Fonte analógica
 $x(t)$

Amostragem
 $x(kT)$

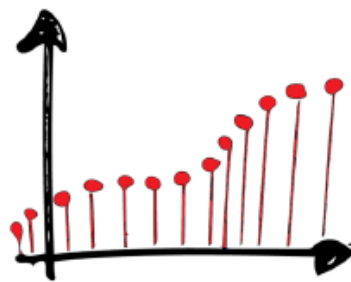
Quantização
 $X_q(k)$

Codificação
 $X_d(k)$

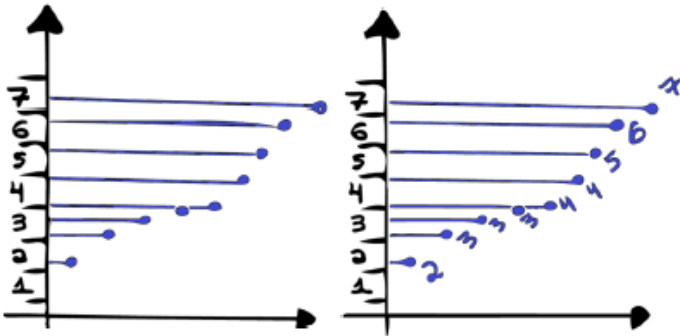
sinal analógico



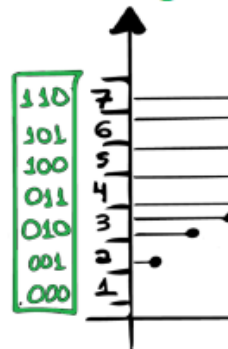
amostrado no tempo



quantizado dentro de um
conjunto de valores inteiros



codificados em
valores digitais

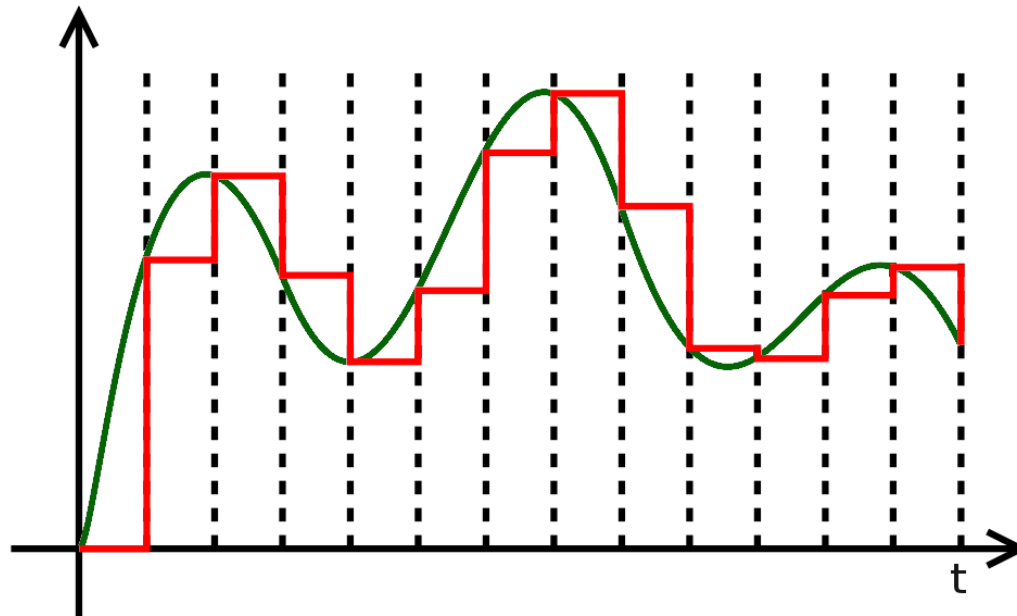


No ATMEGA328 e 2560: 10 bits
 $Q = 2^{10} = 1024$ níveis
Cada nível: $VCC/1024$: 4,88mV

Fonte: Slides DCA0119:
Sistemas Digitais. Prof. Heitor
Medeiros Florencio

Soluções low-cost para AD e processo AD

- A amostragem utiliza técnica de sample-and-hold
- Entre um instante de amostragem e o próximo, mantém valor constante

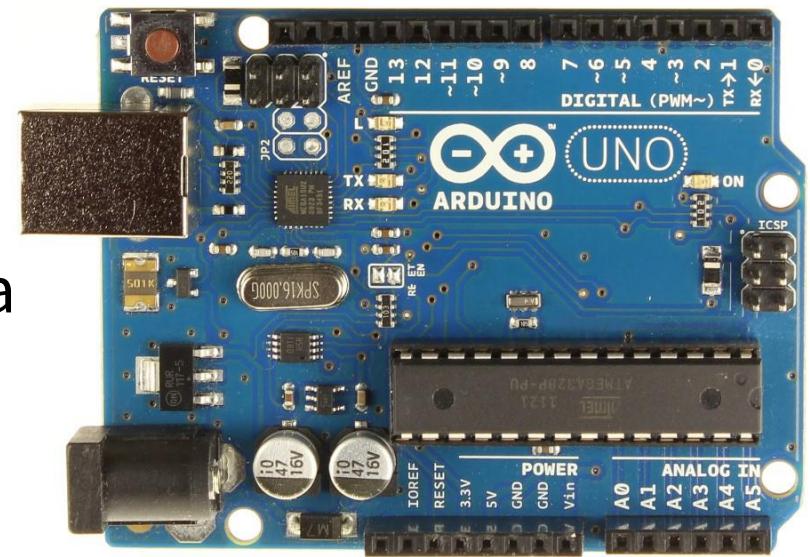


Fonte: <https://www.embarcados.com.br/conversor-a-d/>

Soluções low-cost para AD e processo AD

Fonte: <https://www.embarcados.com.br/conversor-a-d/>

- Configuração padrão do Arduino UNO – ATMEGA328P
- Clock nominal interno: 16 MHz
- Pre-scaler: 128 (divisor padrão)
- Logo: $16.000.000/128 = 125 \text{ kHz}$ (**fADC**)
- Utiliza +-13 ciclos de clock para fazer uma conversão
- Portanto: **+- 9,6 kHz** (frequência máxima de entrada **F_s** a ser amostrado)
- Freq. de amostragem:
 $F_s \geq 2 \times F_{in}$, ou seja,
 $F_{in} \leq F_s/2$ (Frequência de Nyquist) $\leq 4,8 \text{ kHz}$
Ou seja, com este Pre-scaler consegue-se amostrar e posteriormente reconstruir um sinal com até 4,8 kHz de frequência.



Soluções low-cost para AD e processo AD

Faixa de tensão para conversão

- VCC
- Internal (1,1 V)
- Referência (AREF) : max VCC 5V!
- O clock recomendado para o AD em 10 bits é entre 50 e 200 kHz
- Com cristal de $12 \text{ MHz} / 64: 187 \text{ kHz} / 13 = 14384 \text{ amostras/s}$

$$16 \text{ MHz} / 2 = 8 \text{ MHz}$$

$$16 \text{ MHz} / 4 = 4 \text{ MHz}$$

$$16 \text{ MHz} / 8 = 2 \text{ MHz}$$

$$16 \text{ MHz} / 16 = 1 \text{ MHz}$$

$$16 \text{ MHz} / 32 = 500 \text{ kHz}$$

$$16 \text{ MHz} / 64 = 250 \text{ kHz}$$

$$16 \text{ MHz} / 128 = 125 \text{ kHz}$$

Fonte: <https://www.embarcados.com.br/conversor-a-d/>

Soluções low-cost para AD e processo AD

Existem outros conversores

- ADS1115 (16 bits) e ADS1015 (12 bits): I2C (SDA, SCL)
- No PIC 18F4550: 10 bits
- No ESP8266/nodeMCU: 01 canal A0 de 10 bits
- Pode expandir com MCP3008 (SPI): 08 canais 10 bits ou MCP3208 de 12 bits e 08 canais (SPI): MOSI/MISO



MCP3008 - Conversor Analógico/Digital

Código: MCP3008DIP

R\$ 21,90

até 3x de **R\$ 7,30** sem juros
ou **R\$ 20,80** via depósito

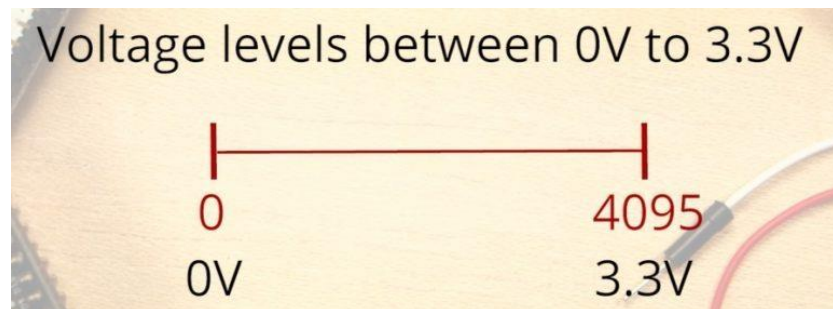
Autocore robotica

Soluções low-cost para AD e processo AD



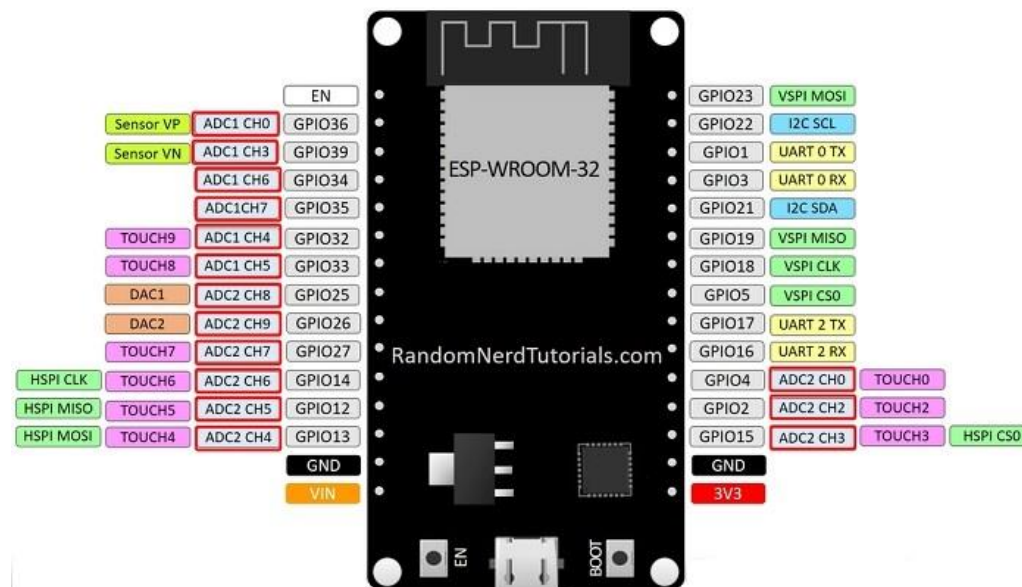
No ESP32: 15 canais

- Porta 1 (ADC1) com 06 canais (CH0,CH3,CH4,CH5,CH6,CH7)
- Porta 2 (ADC2) com 09 canais (CH0,CH2,CH3,...,CH9)
- Padrão de resolução: 12 bits (0-4095)
- Ciclos por conversão no ESP32, padrão 8
- <https://randomnerdtutorials.com/esp32-adc-analog-read-arduino-ide/>



Especificações	ESP8266	ESP32
MCU	xtensa® single 32-bit 1106	xtensa® dual-core 32-bit lx6 600 DMIPS
802.11 b/g/n Wi-Fi	HT20	HT40
Bluetooth	Não	Bluetooth 4.2 Le
Frequência	80 MHz	160 MHz
SRAM	160 Kbytes	512 Kbytes
Flash	SPI Flash, 16 Mbytes	SPI Flash, 16 Mbytes
GPIO	17	36
Hardware/ Software PWM	Não/ 8 canais	1 / 16 canais
SPI/ I2C/I2S/UART	2/1/2/2	4/2/2/2
ADC	10-bit	12-bit
CAN	Não	1
Interface Ethernet Mac	Não	1
Sensor Capacitivo	Não	Sim
Sensor de Temperatura	Não	Sim
Temperatura de Trabalho	-40 °C a 125 °C	-40 °C a 125 °C

ESP32 DEVKIT V1 - DOIT



Soluções low-cost para AD e processo AD

Exemplo de código para medir tempo de conversão Configuração padrão

- Entre 112us e 116us
- Com scaler de 64,
- 60 us (dobro freq)
- em torno de 16kHz

```
unsigned long tempo_inicio;
unsigned long tempo_fim;
unsigned long valor;

void setup() {
  Serial.begin(115200);    //inicia a comunicação serial
}

void loop() {
  // leitura
  tempo_inicio = micros(); //marca tempo de inicio de leitura
  valor = analogRead(0);   //le valor convertido
  tempo_fim = micros();    //le tempo no fim da conversão

  //exibe valor lido e tempo de conversão
  Serial.print("Valor = ");
  Serial.print(valor);
  Serial.print(" -- Tempo leitura = ");
  Serial.print(tempo_fim - tempo_inicio);
  Serial.println(" us");
  delay(500);
}
```


Soluções low-cost para AD e processo AD

Em testes no embarcados.com.br, conseguiu manter os 10 bits até 1 Mhz (pre scaler 16), com Fs de 50kHz!

ADCSRA – ADC Control and Status Register A

Bit	7	6	5	4	3	2	1	0	
(0x7A)	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
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	

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

Configura o fator de divisão entre o clock do sistema e a entrada de clock do ADC. Os valores possíveis são exibidos na tabela abaixo:

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