5197 - Sistema Digitais

Bacharelado de Informática

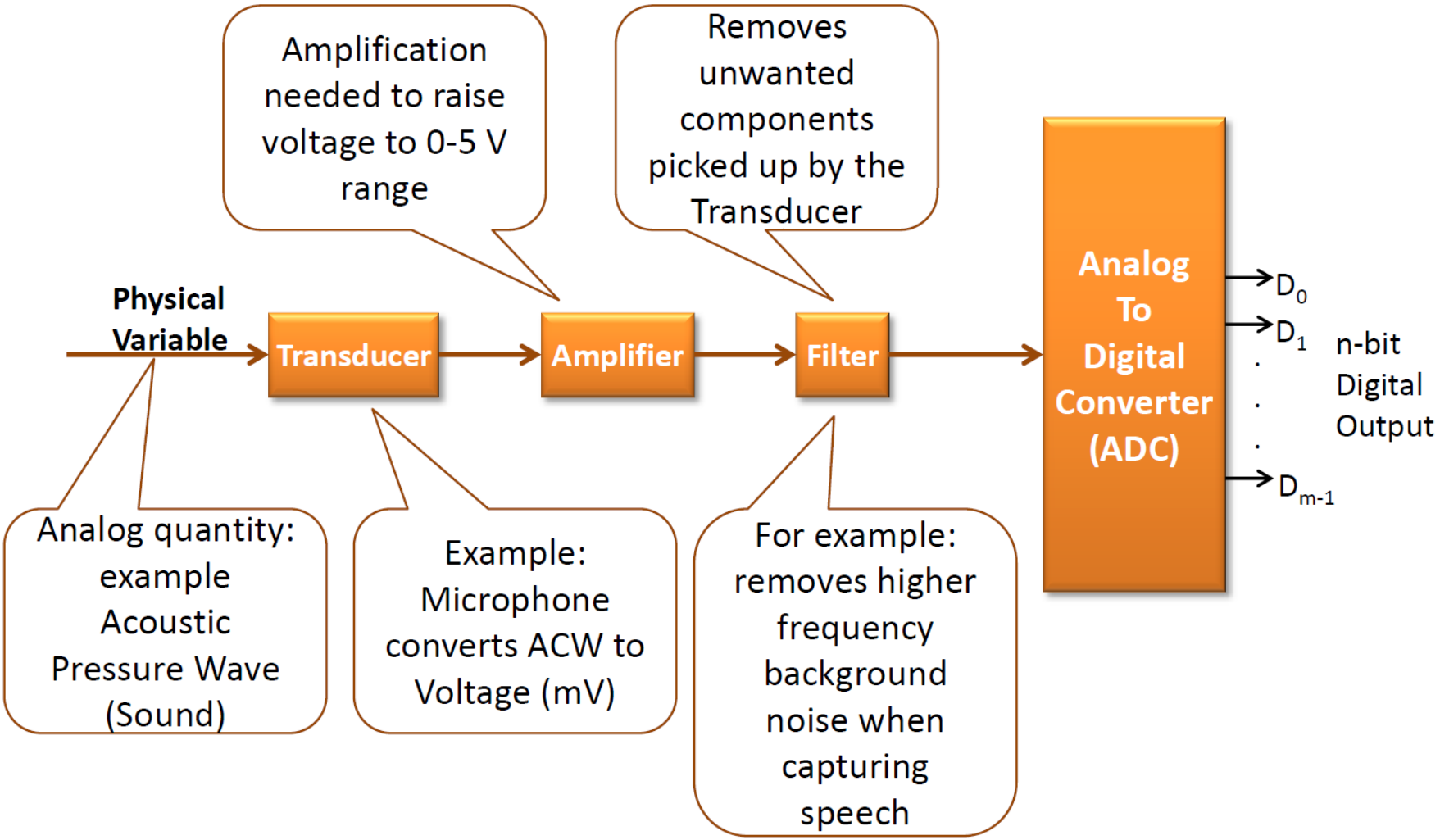
UEM – DIN - Prof. Elvio

2016

Roteiro

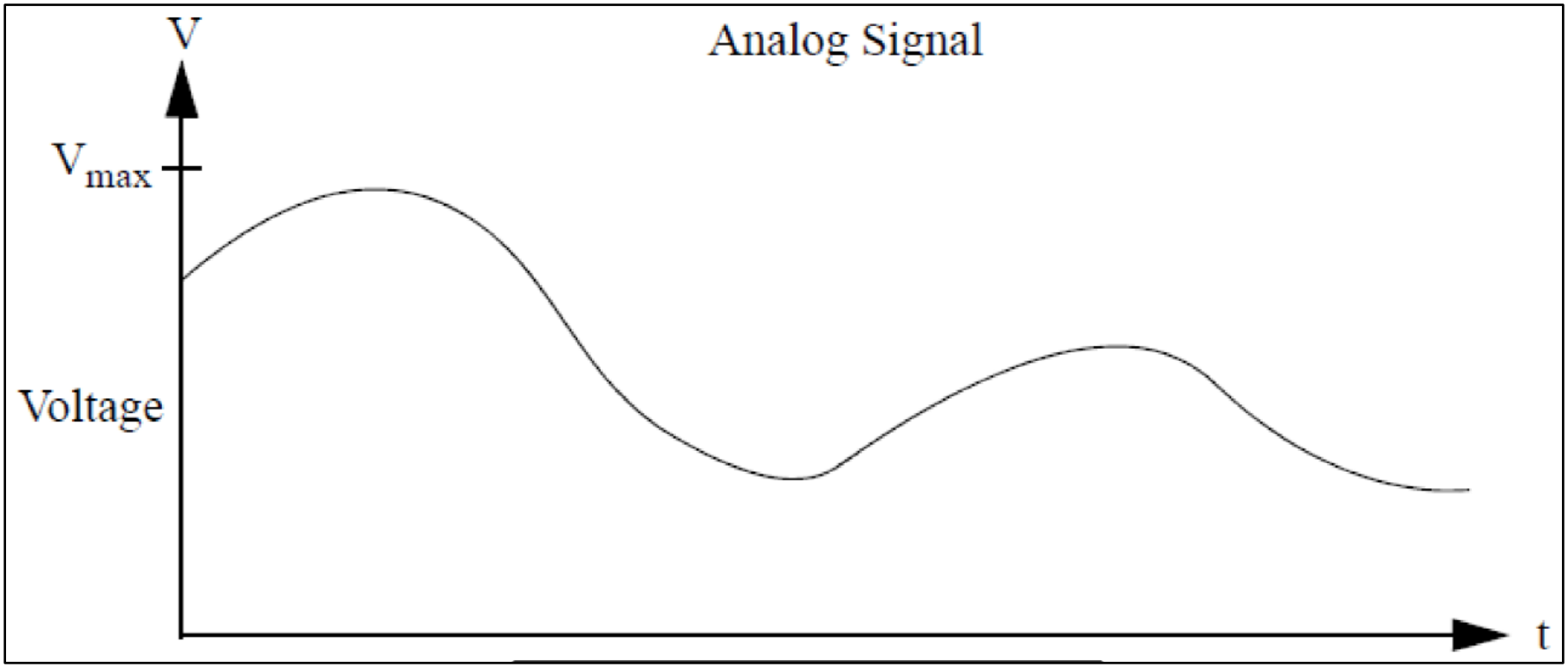
- Conversor Analógico/Digital

Aquisição de Dados



Sinal Analógico

- Sinal contínuo na ordenada e na abscissa

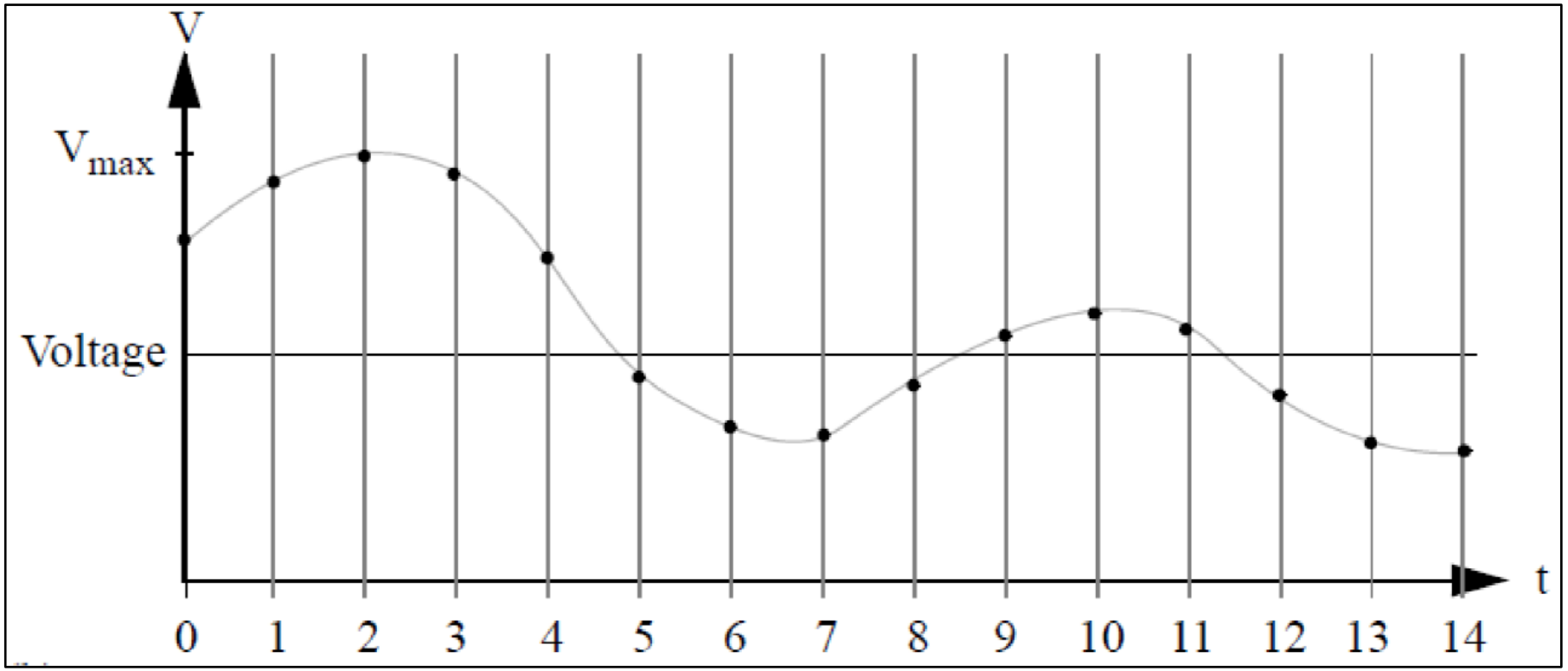


Digitalização

- Processo de conversão de sinal analógico em digital
- Envolve a amostragem do sinal analógico em pontos discretos no tempo, usualmente em intervalos periódicos
- Para cada amostra feita, aproxima-se o valor para o valor discreto mais próximo

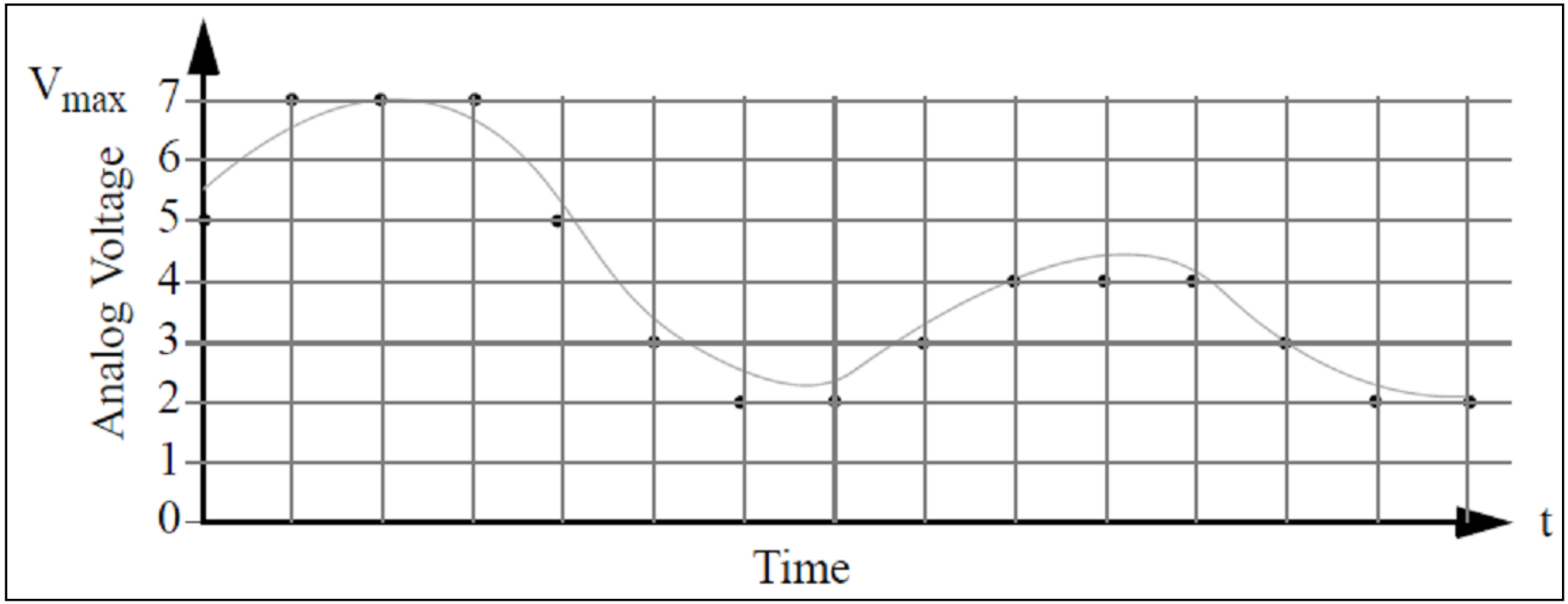
Digitalização

- **Amostragem**: sinal é amostrado em instantes específicos no tempo



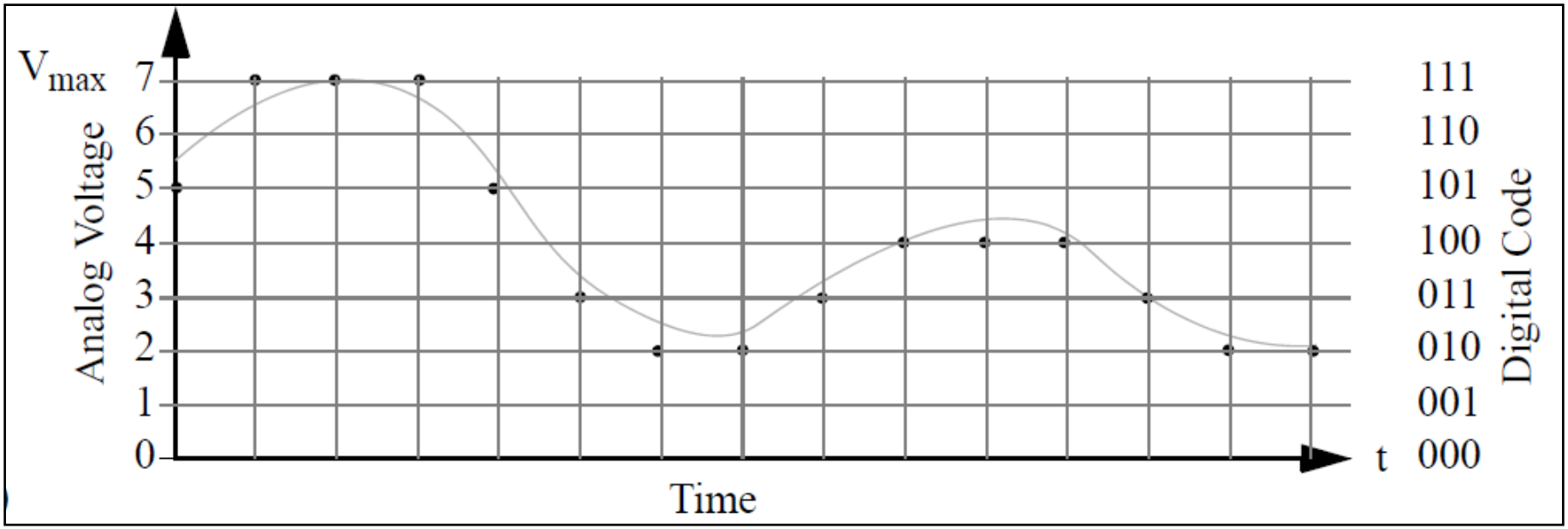
Digitalização

- **Quantização**: valor amostrado é aproximado para o valor discreto mais próximo



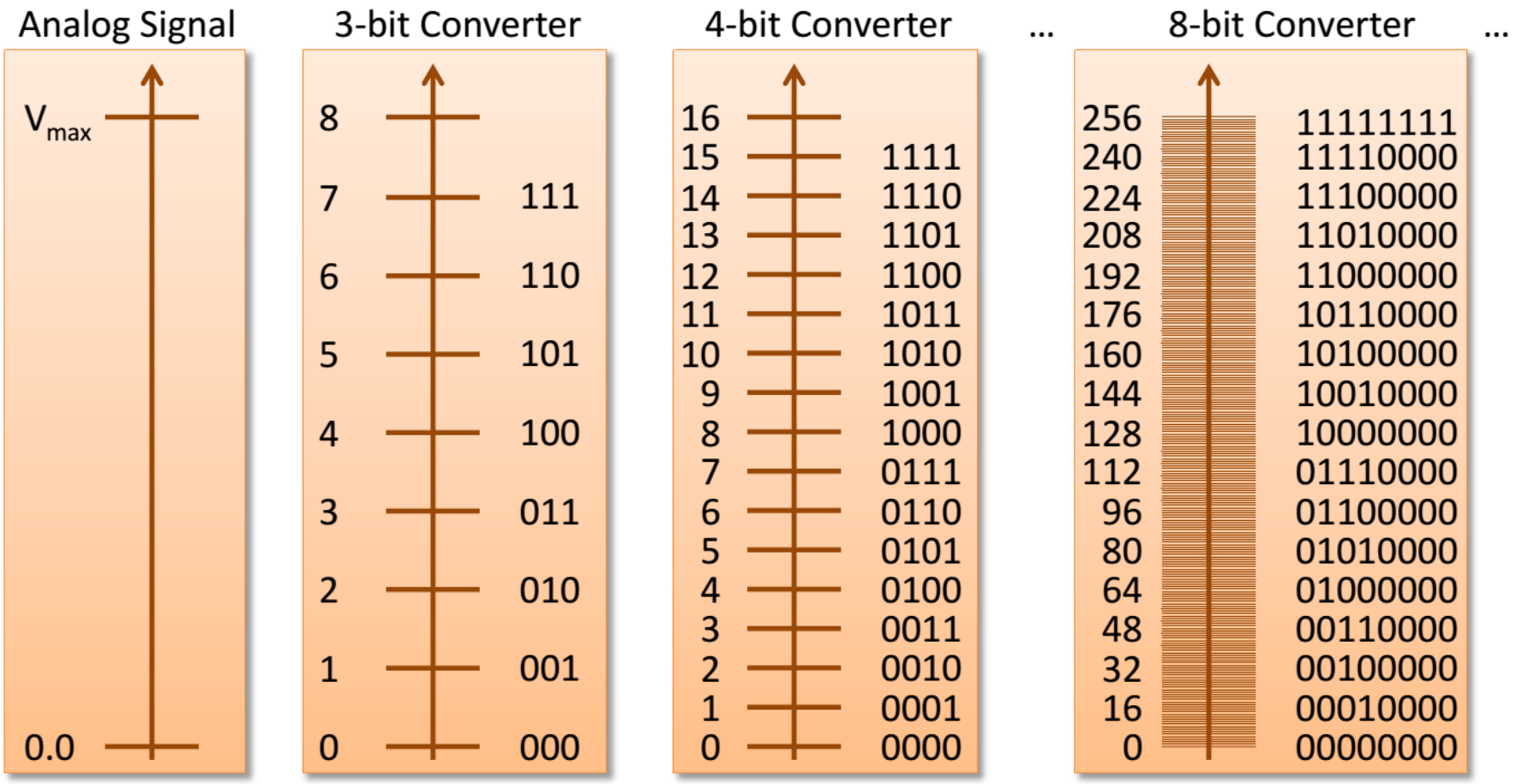
Digitalização

- **Representação Digital**: cada nível discreto recebe um valor digital



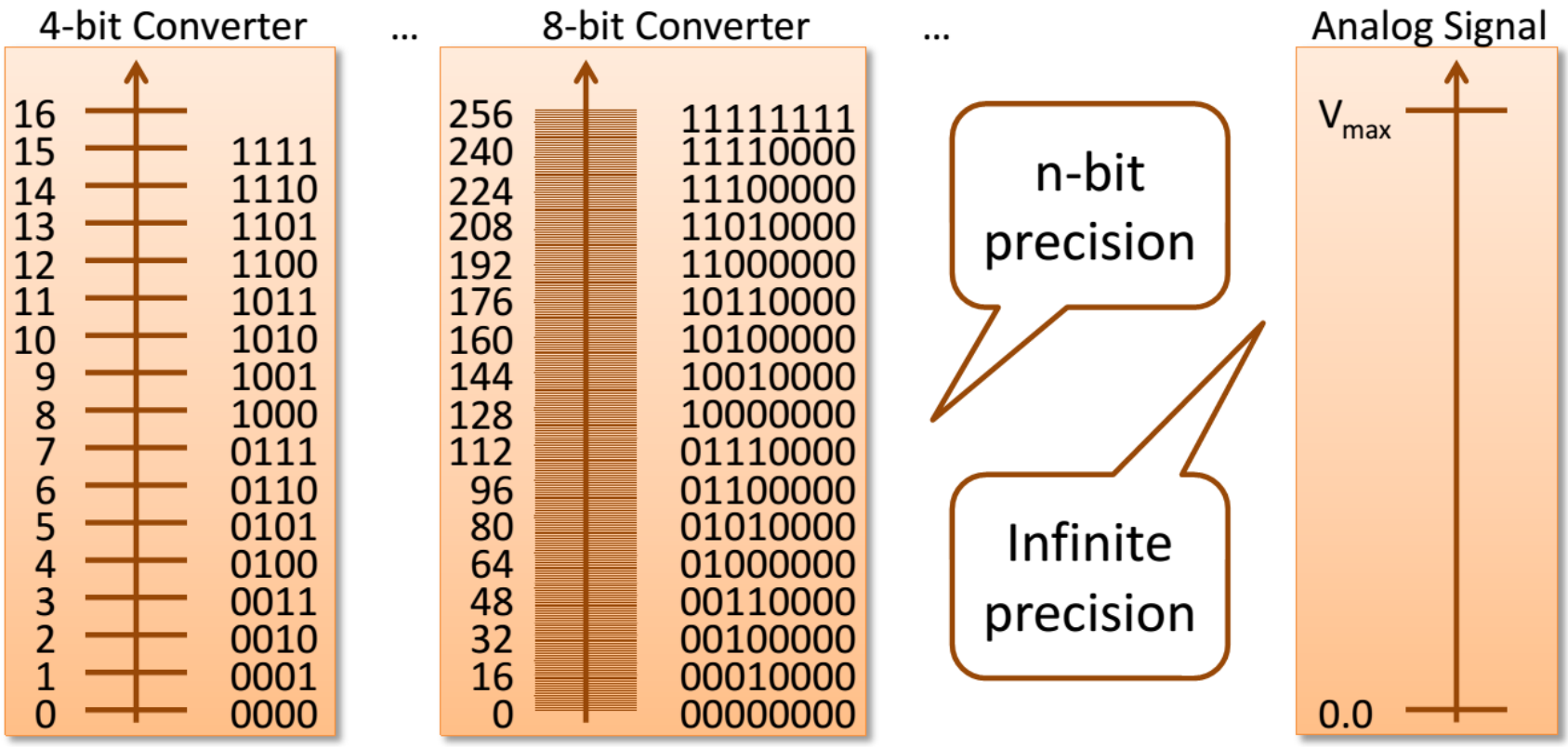
Digitalização

- Quantos mais bits, mais níveis discretos



Digitalização

- Quantos mais níveis discretos, menor o **erro de quantização**

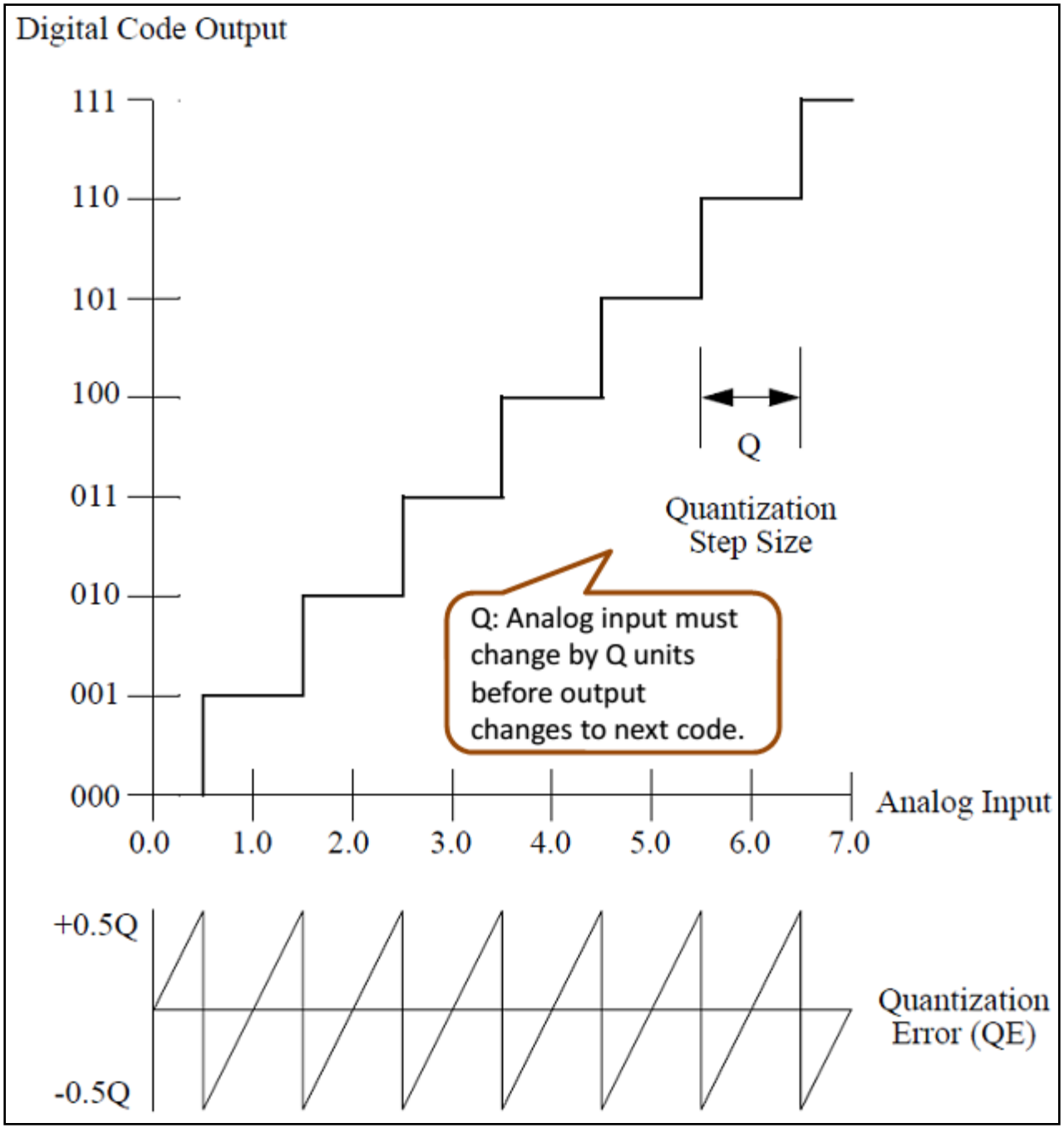


Digitalização

- Erro de quantização não pode ser recuperado
- Tamanho do erro de quantização diminui com o aumento no número de bits utilizados
- Quanto erro é aceitável?
  - Depende da aplicação
- Aplicação determina o número de bits necessários
  - Música: 16 bits
  - Voz (qualidade de telefone): 8 bits

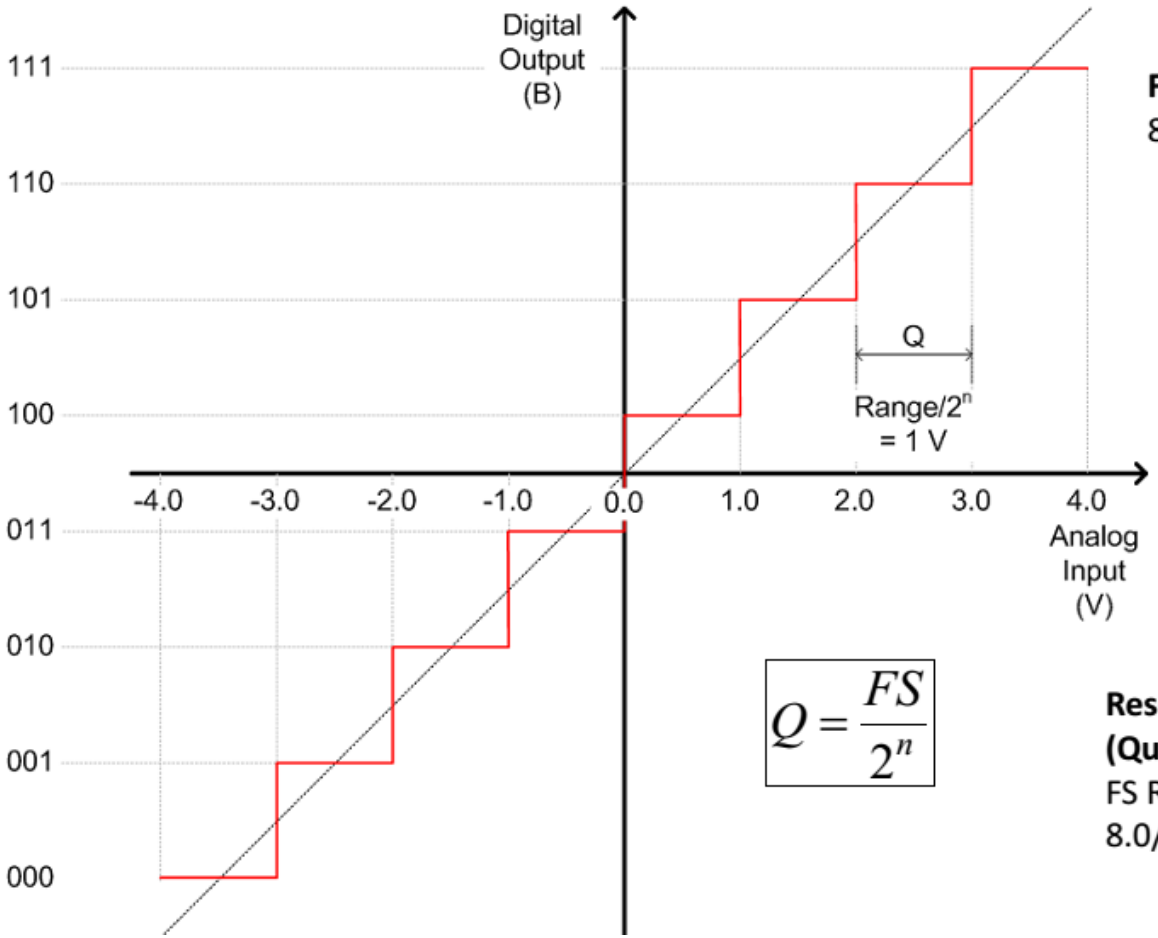
Digitalização

- Erro de quantização



Digitalização

- Exemplo: 3 bits bipolar



Full Scale (FS) Range:
8.0 V

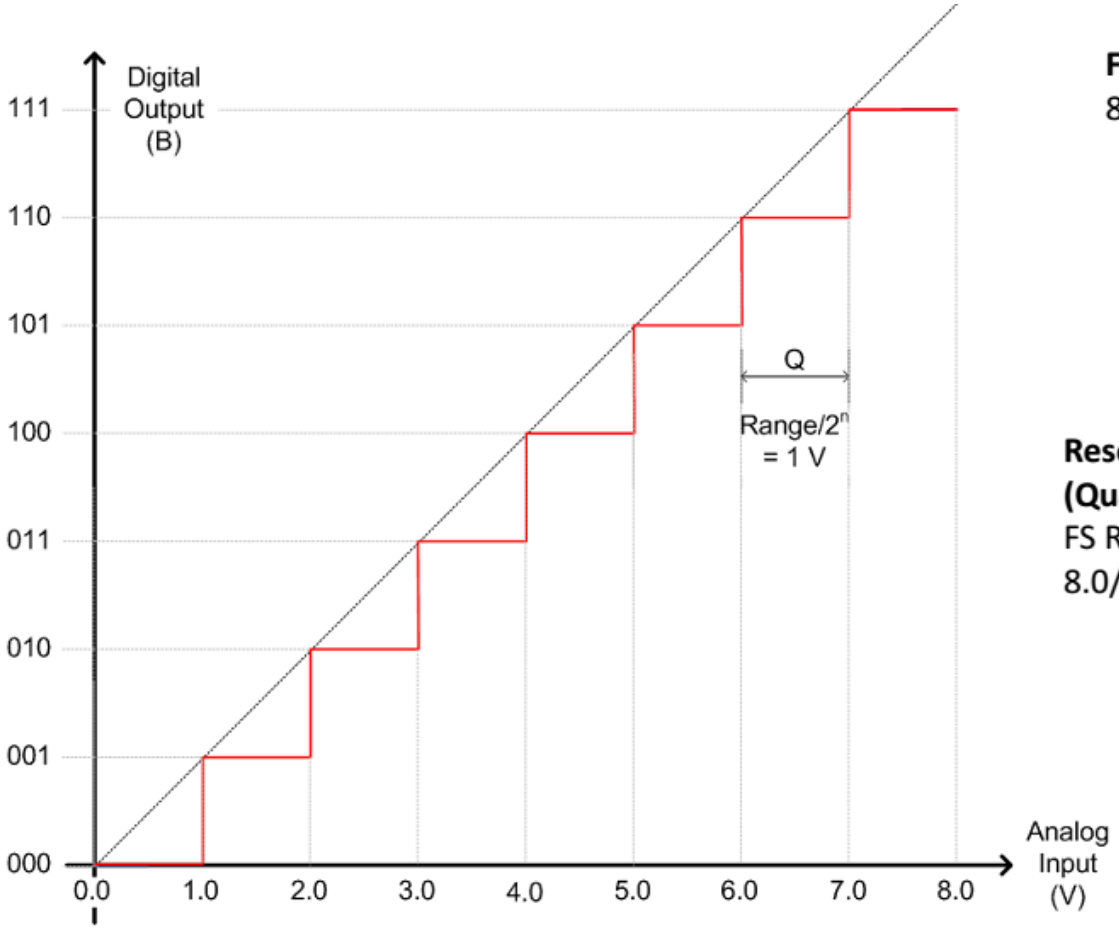
Precision:
3-Bit,
8 Possibilities

$$Q = \frac{FS}{2^n}$$

Resolution (Quantization Step Size, Q):
FS Range/Number of Steps
8.0/8 = 1.0 V

Digitalização

- Exemplo: 3 bits unipolar



Full Scale (FS) Range:
8.0 V

Precision:
3-Bit,
8 Possibilities

Resolution
(Quantization Step Size, Q):
FS Range/Number of Steps
8.0/8 = 1.0 V

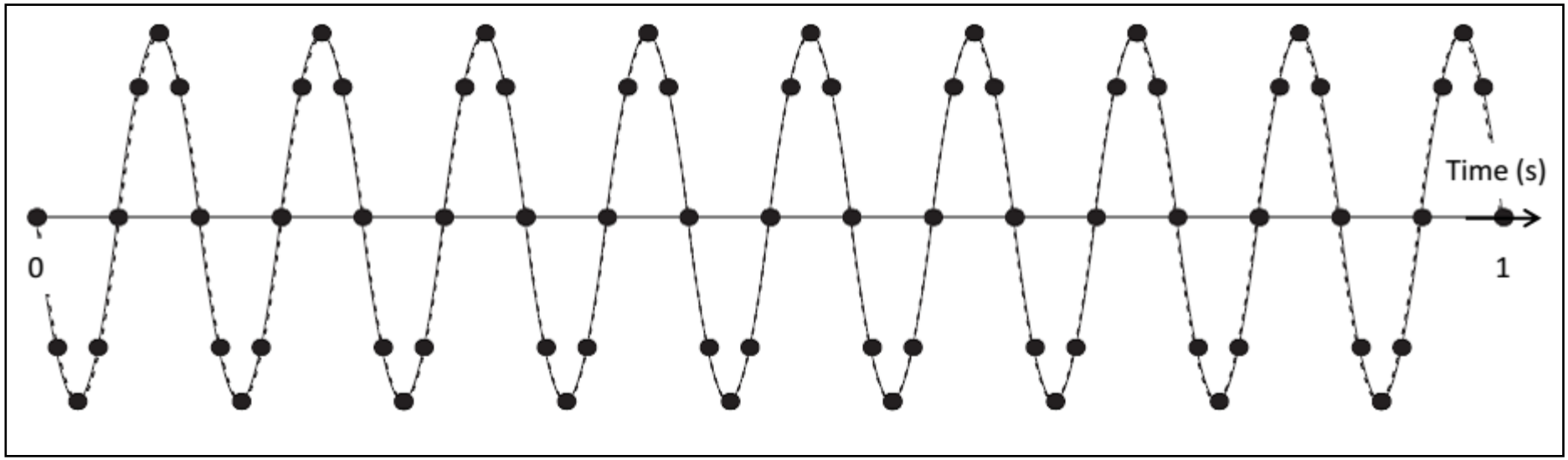
$$Q = \frac{FS}{2^n}$$

Frequência de Amostragem

- Qual deve ser a taxa de amostragem mínima para produzir uma representação digital precisa do sinal, de maneira que ele possa ser reconstruido a partir do sinal amostrado?
- Intuitivamente, espera-se que a taxa de amostragem tenha relação com a taxa com que o sinal mude
  - Pouca mudança do sinal: taxa de amostragem mais baixa
  - Mudanças rápidas do sinal: taxa de amostragem mais alta

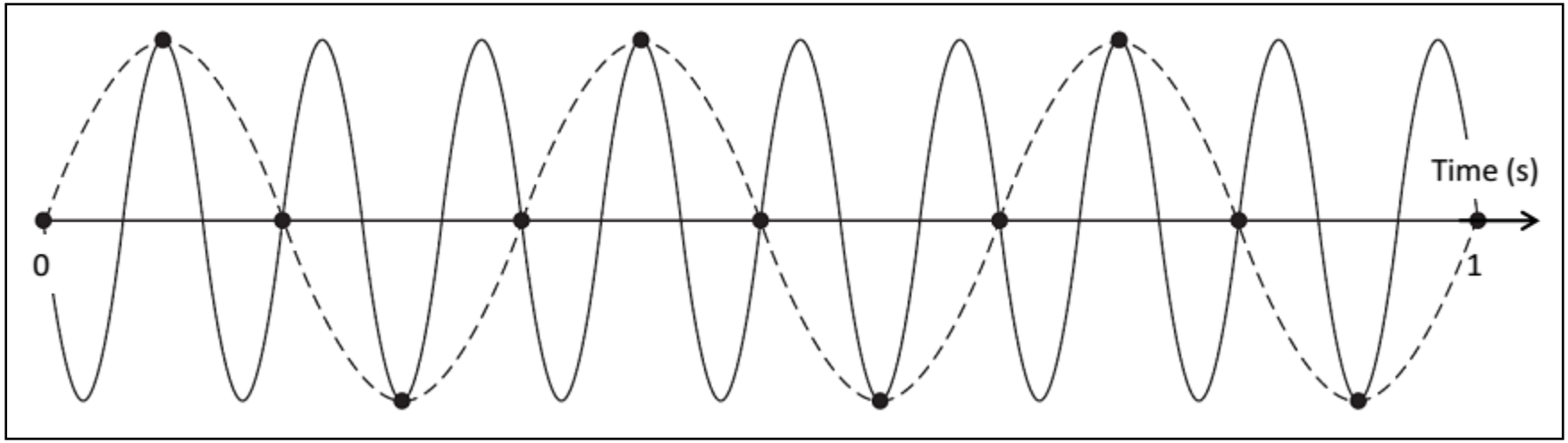
Frequência de Amostragem

- Exemplo: senóide com frequência de 9 Hz
  - Amostras por período: 8 amostras
  - Frequência de amostragem: 8 x 9 = 72 a/s



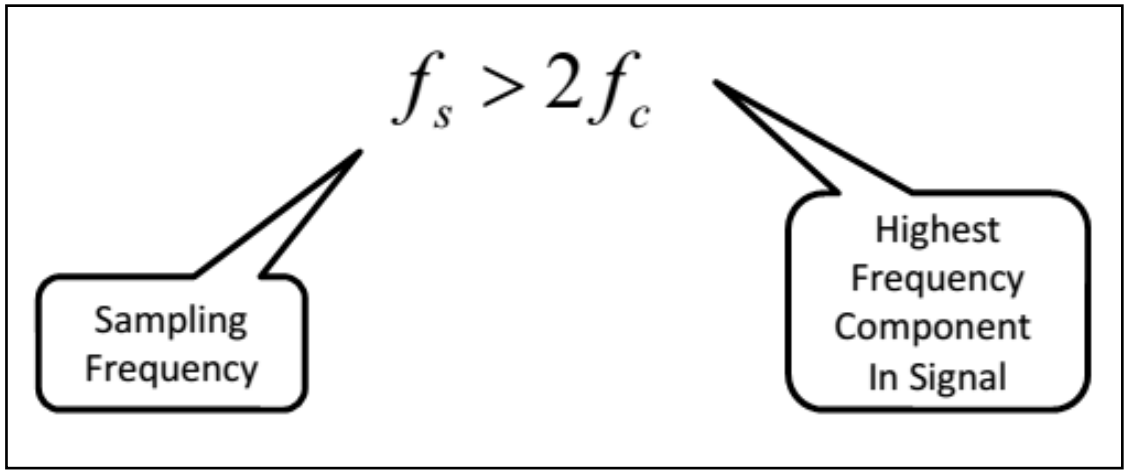
Frequência de Amostragem

- Exemplo: senóide com frequência de 9 Hz
  - Frequência de amostragem: 12 a/s
  - Sinal recuperado a partir das amostras parece ter apenas 3 Hz
  - **Aliasing**: erro causado por frequência de amostragem muito baixa



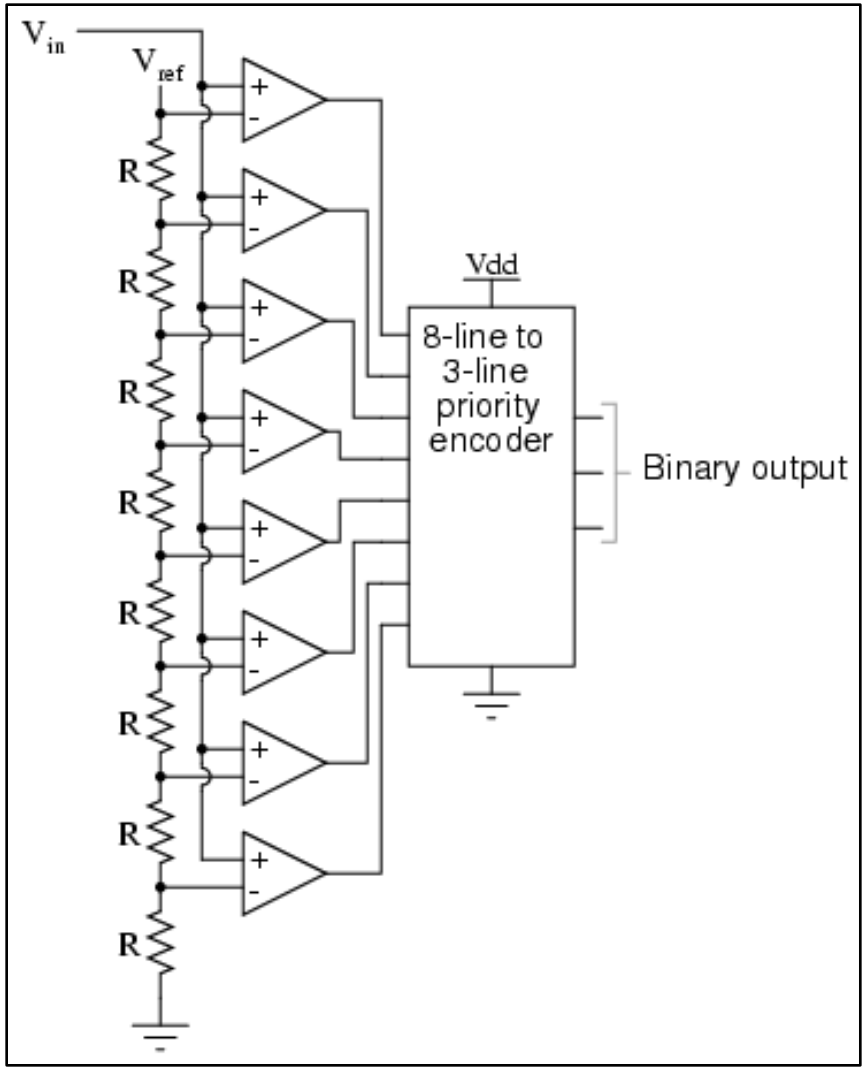
Frequência de Amostragem

- Teorema de Nyquist-Shannon
  - Se um sinal contínuo contendo componentes com frequência máxima igual a f_c for amostrado a uma frequência maior que $2 f_c$, o sinal original pode ser completamente recuperado a partir das amostras sem outro erro além daquele introduzido pela quantização.



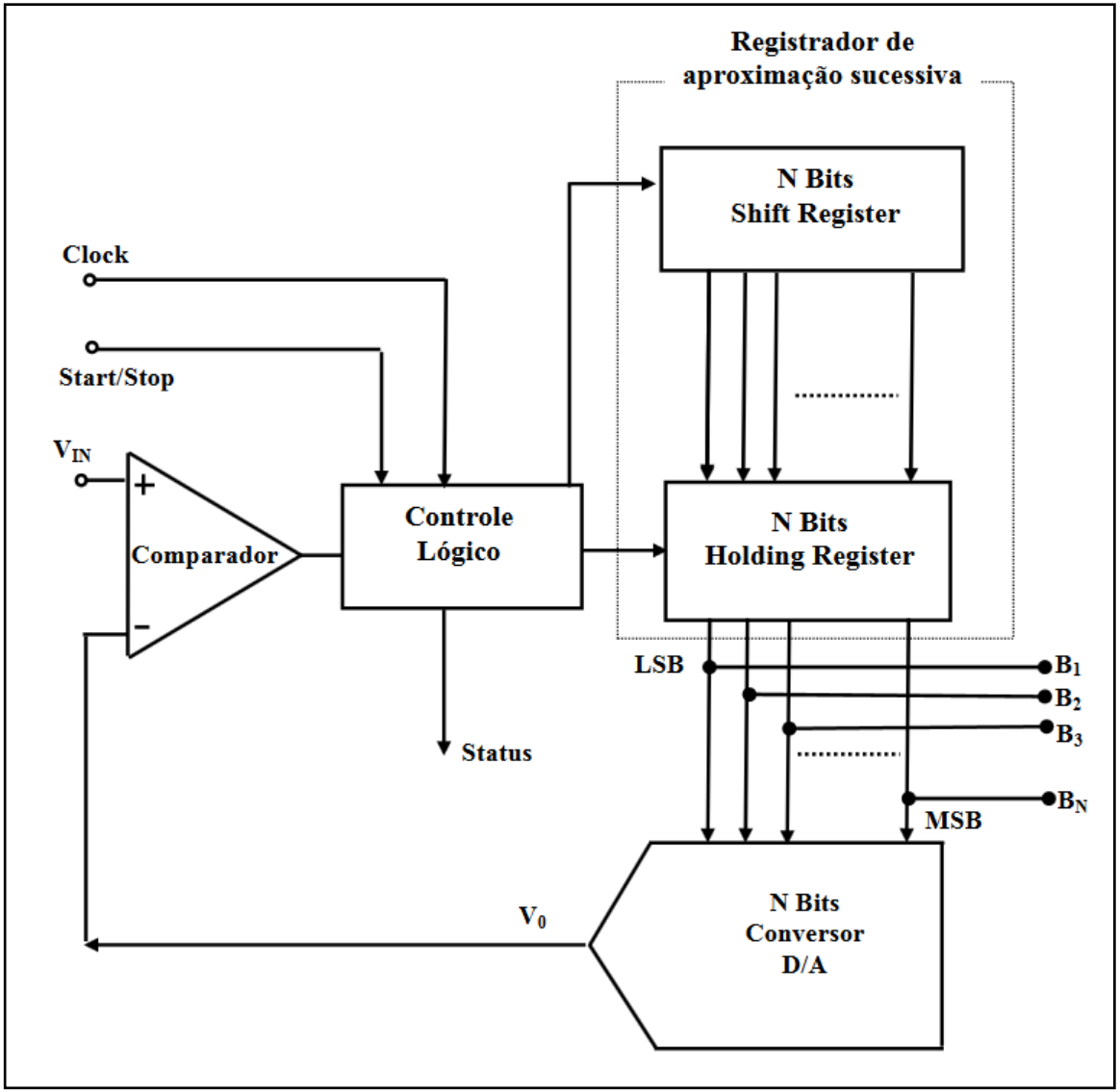
Tipos de Conversores A/D

- Flash
  - Operação em paralelo, através de um conjunto de comparadores
  - Cada comparador compara o sinal de entrada com uma fração do valor de referência, dependendo do número de bits da conversão
  - **Vantagem**: rápido
  - **Desvantagens**: caro, grande e consume muita energia



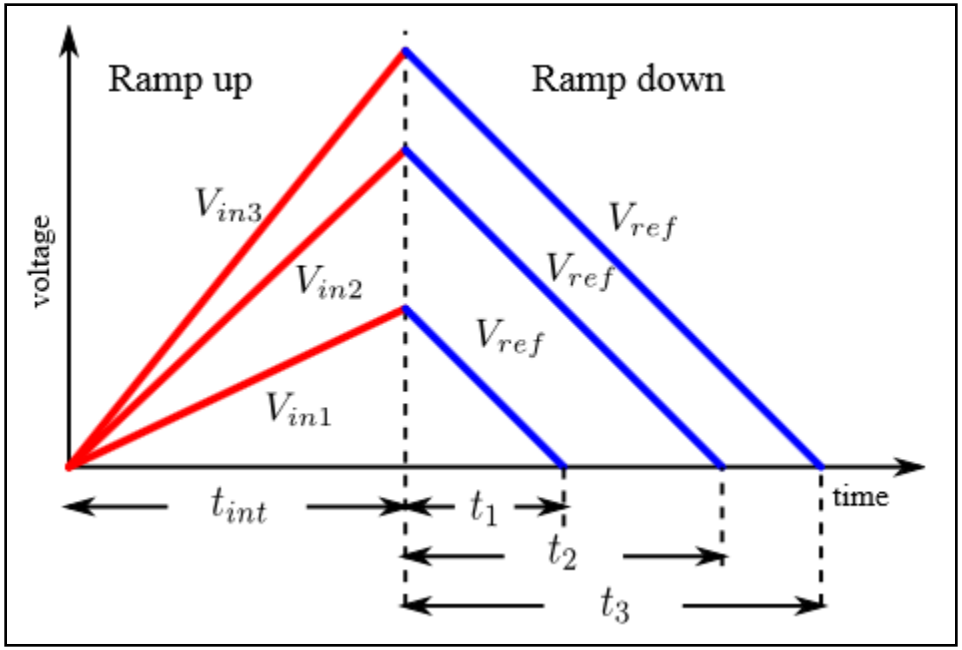
Tipos de Conversores A/D

- Aproximação Sucessiva
  - Utiliza um Conversor Digital/Analógico para verificar acerto na conversão
  - **Vantagens**: rápido e com boa precisão
  - **Desvantagem**: velocidade cai com aumento na precisão



Tipos de Conversores A/D

- Rampa
  - Rampa de subida proporcional ao valor do tensão de entrada
  - Rampa de descida com inclinação constante
  - Tempo de descida proporcional ao valor do tensão de entrada
  - **Vantagens**: alta precisão e imunidade a ruído
  - **Desvantagem**: lento



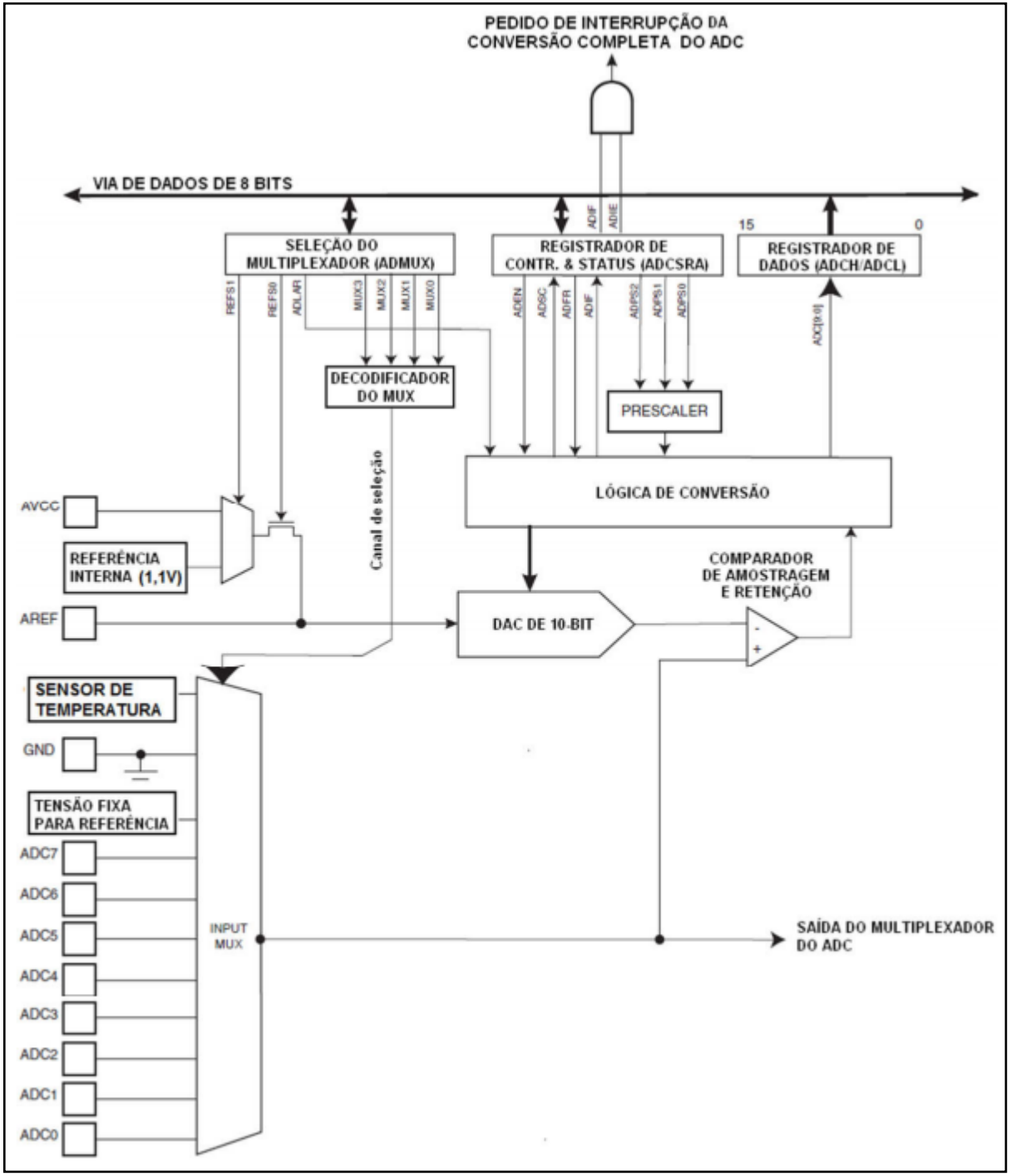
$$t = \frac{V_{in}}{V_{ref}} t_{int}$$

Conversor A/D do Arduino

- Características principais
  - Conversor com aproximação sucessiva
  - 10 bits de resolução (1024 pontos).
  - Tempo de conversão de 13 a 260 µs
  - Até 76,9 k amostras/segundo, 15 k a/s na resolução máxima.
  - 6 canais de entrada multiplexados
  - Faixa de tensão de entrada de 0 a V_{CC}
  - Tensão de referência selecionável de 1,1 V
  - Modo de conversão simples ou contínua
  - Interrupção ao término da conversão
  - Eliminador de ruído para o modo Sleep
  - Sensor interno de temperatura com ± 10 ºC de precisão

Conversor A/D do Arduino

- Diagrama em blocos

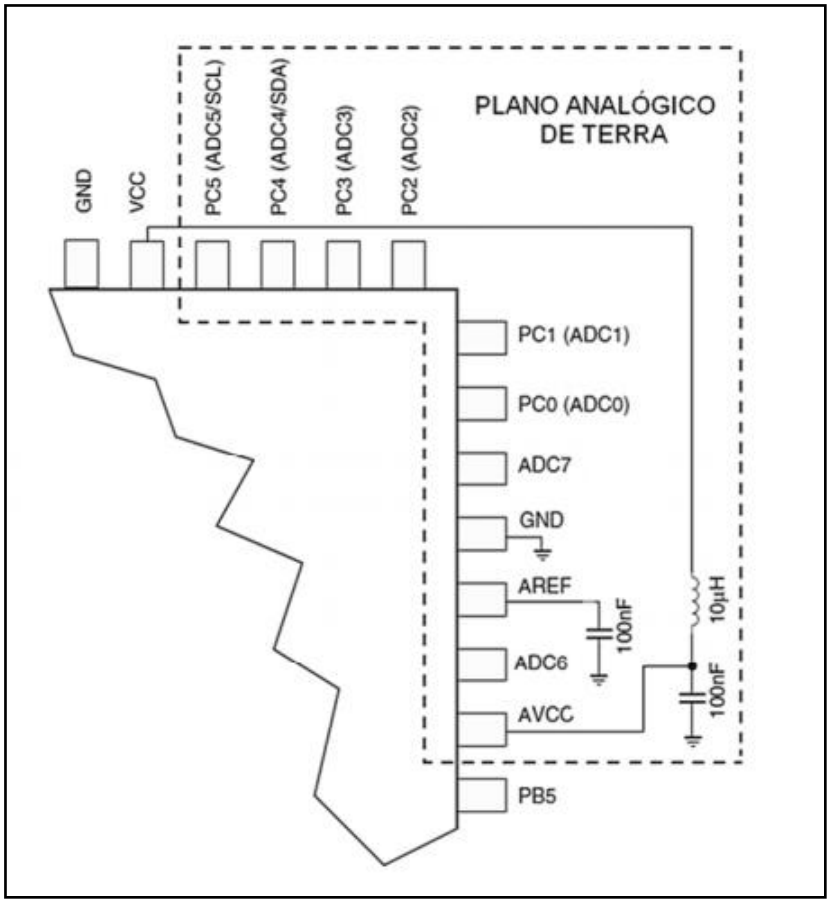


Conversor A/D do Arduino

- Valor convertido:

$$\text{ADC} = \frac{V_{IN}.1024}{V_{REF}}$$

- Circuito recomendado



Conversor A/D do Arduino

- Tempos de conversão

Condition	Sample & Hold (Cycles from Start of Conversion)	Conversion Time (Cycles)
First conversion	13.5	25
Normal conversions, single ended	1.5	13
Auto Triggered conversions	2	13.5

- Registrador *ADC Multiplexer Selection* (ADMUX)

Bit	7	6	5	4	3	2	1	0
(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

Conversor A/D do Arduino

- Registrador *ADC Multiplexer Selection* (ADMUX)
  - *Reference Selection* (REFS[1:0])
    - Seleciona a tensão de referência

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

  - *ADC Left Adjust Result* (ADLAR)
    - Altera como o resultado é apresentado (veja figura adiante)

Conversor A/D do Arduino

- Registrador *ADC Multiplexer Selection* (ADMUX)
  - *ADC Left Adjust Result* (ADLAR)

		15	14	13	12	11	10	9	8
ADLAR=0	**ADCH**	-	-	-	-	-	-	ADC9	ADC8
	ADCL	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0
Bit		7	6	5	4	3	2	1	0
Lê/Escreve		L	L	L	L	L	L	L	L
		L	L	L	L	L	L	L	L
Valor Inicial		0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0

		15	14	13	12	11	10	9	8
ADLAR=1	**ADCH**	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2
	ADCL	ADC1	ADC0	-	-	-	-	-	-
Bit		7	6	5	4	3	2	1	0
Lê/Escreve		L	L	L	L	L	L	L	L
		L	L	L	L	L	L	L	L
Valor Inicial		0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0

Conversor A/D do Arduino

- Registrador *ADC Multiplexer Selection* (ADMUX)
  - *Analog Channel Selection* (MUX[3:0])

MUX3...0	Single Ended Input
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4
0101	ADC5
0110	ADC6
0111	ADC7
1000	ADC8(1)
1001	(reserved)
1010	(reserved)
1011	(reserved)
1100	(reserved)
1101	(reserved)
1110	1.1V (V_{BG})
1111	0V (GND)

Conversor A/D do Arduino

- Reg. *ADC Control and Status A* (ADCSRA)

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

  - *ADC Enable* (ADEN)
    - Habilita o conversor
  - *ADC Start Conversion* (ADSC)
    - Inicia a conversão A/D
  - *ADC Auto Trigger Enable* (ADATE)
    - Inicia a conversão com o gatilho designado no ADCSRB
  - *ADC Interrupt Flag* (ADIF)
    - Indica que a conversão foi finalizada e o resultado está disponível para leitura

Conversor A/D do Arduino

- Reg. *ADC Control and Status A* (ADCSRA)
  - *ADC Interrupt Enable* (ADIE)
    - Habita a interrupção causada pelo fim da conversão
  - *ADC Prescaler Select* (ADPS[2:0])
    - Seleciona o relógio do conversor

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

Conversor A/D do Arduino

- Reg. *ADC Control and Status B* (ADCSRB)

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

  - *ADC Auto Trigger Source* (ADTS[2:0])
    - Seleciona a fonte do autodisparo do conversor
  - *Analog Comparator Multiplexer Enabled* (ACME)

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

Conversor A/D do Arduino

- Registradores *ADC Data* (ADCL e ADCH)
  - *ADC Conversion Result* (ADC9:0)
- Registrador *Digital Input Disable* 0 (DIDR0)

Bit	7	6	5	4	3	2	1	0
(0x7E)	–	–	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

  - *ADC5-0 Digital Input Disable* (ADC5D-ADC0D)
    - Desabilita a entrada analógica correspondente

Conversor A/D do Arduino

- Exemplo de código

```
int main(void)
{
	USART_Inic(MYUBRR);

	//Tensão interna de ref (1.1V), sensor de temp. do chip
	ADMUX = (1<<REFS1) | (1<<REFS0) | (1<<MUX3);

	//habilita o AD e define um prescaler de 128 (clk_AD = F_CPU/128), 125 kHz
	ADCSRA = (1<<ADEN) | (1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);
	escreve_USART_Flash(msg);

	while(1)
	{
		ident_num((unsigned int)le_temp(),digitos);//leitura de temperatura sem sinal
		USART_Transmite(digitos[2]);
		USART_Transmite(digitos[1]);
		USART_Transmite(digitos[0]);
		USART_Transmite(176);	//símbolo 'o'
		USART_Transmite('C');
		USART_Transmite('\n');	//nova linha
		_delay_ms(1000);
	}
}
//-----------------------------------------------------------------------------
signed int le_temp()
{
	set_bit(ADCSRA, ADSC);		//inicia a conversão
	while(tst_bit(ADCSRA,ADSC));	//espera a conversão ser finalizada

	return (ADC - Offset_temp);	//fator k de divisão = 1
}
```