

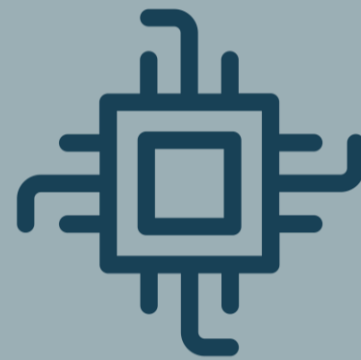


Universidade Federal
de Campina Grande



ADC

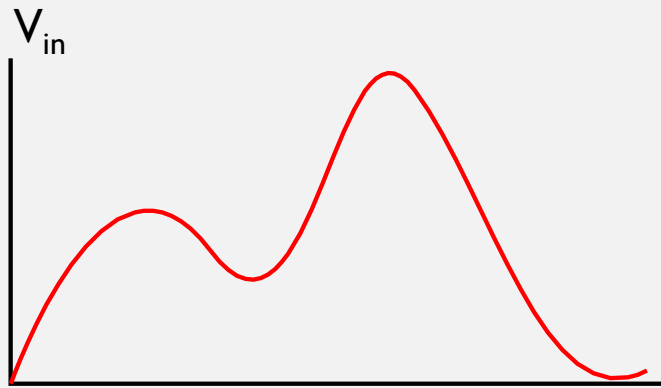
ATMEGA328P



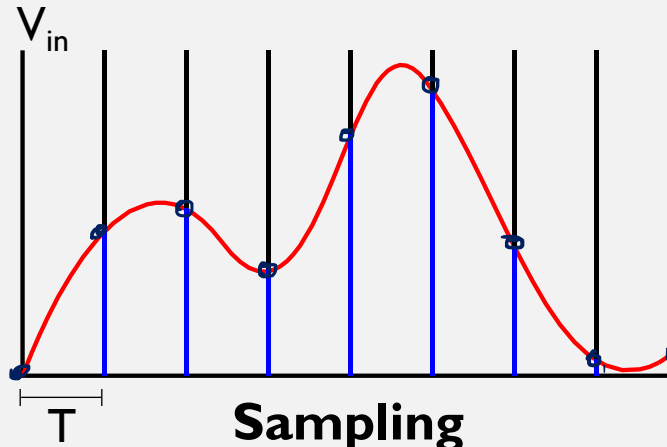
Prof.

**Rafael
Lima**

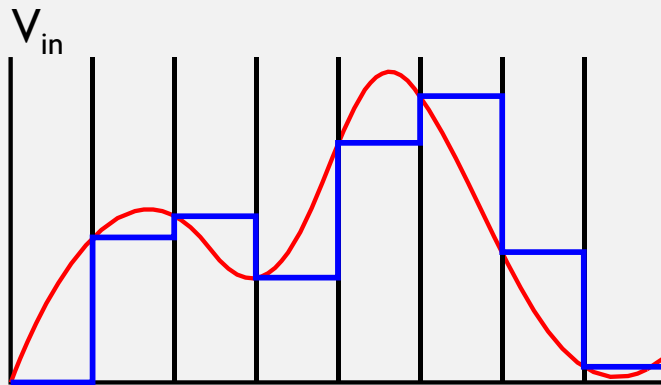
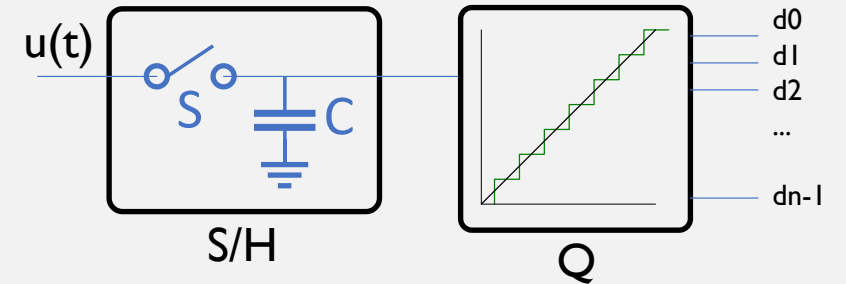
CONVERSÃO ANALÓGICO-DIGITAL



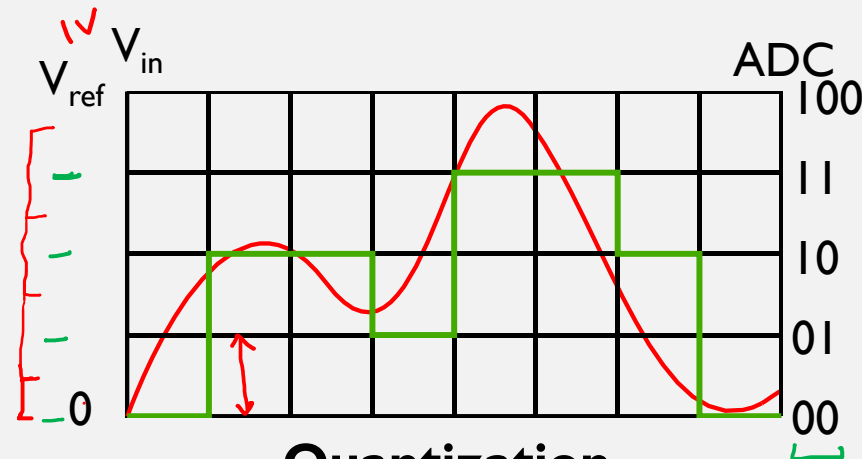
Sinal Analógico



Sampling



Sampling and Hold



Quantization

$$ADC = (V_{in}/V_{ref}) 2^n$$

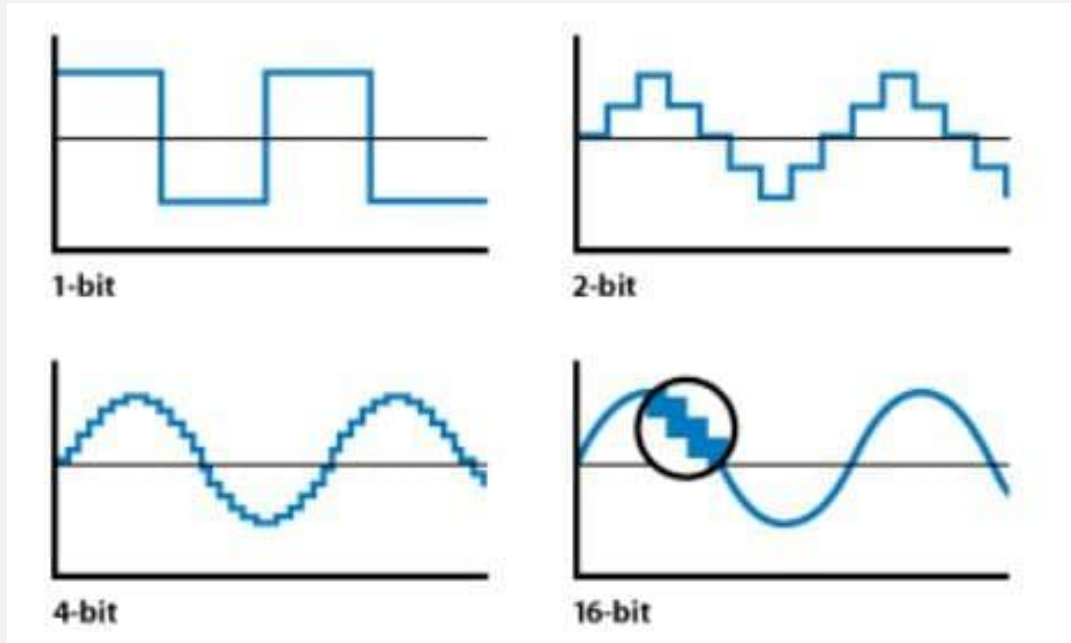
$$Resl. = V_{ref} / 2^n$$

Handwritten notes in red:

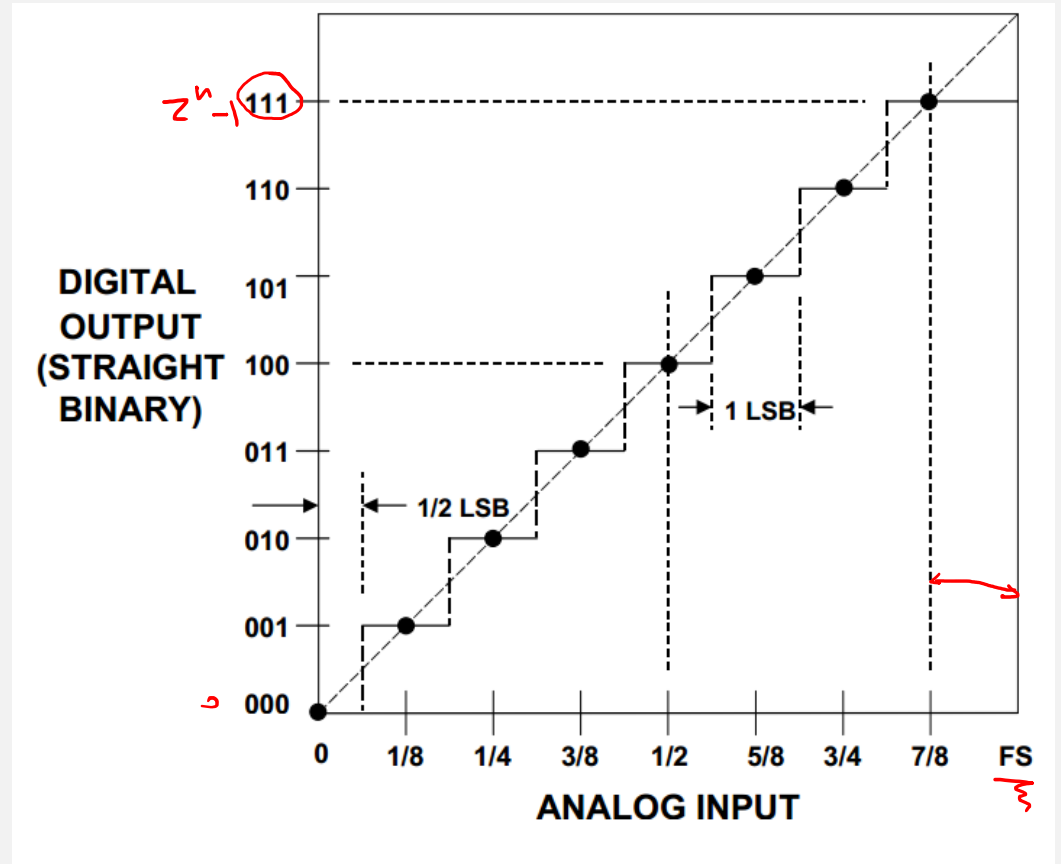
$$ADC = \frac{V_{in}}{1} \cdot 4$$

$$Resl. = \frac{1}{4} = 0,25V$$

CONVERSÃO ANALÓGICO-DIGITAL

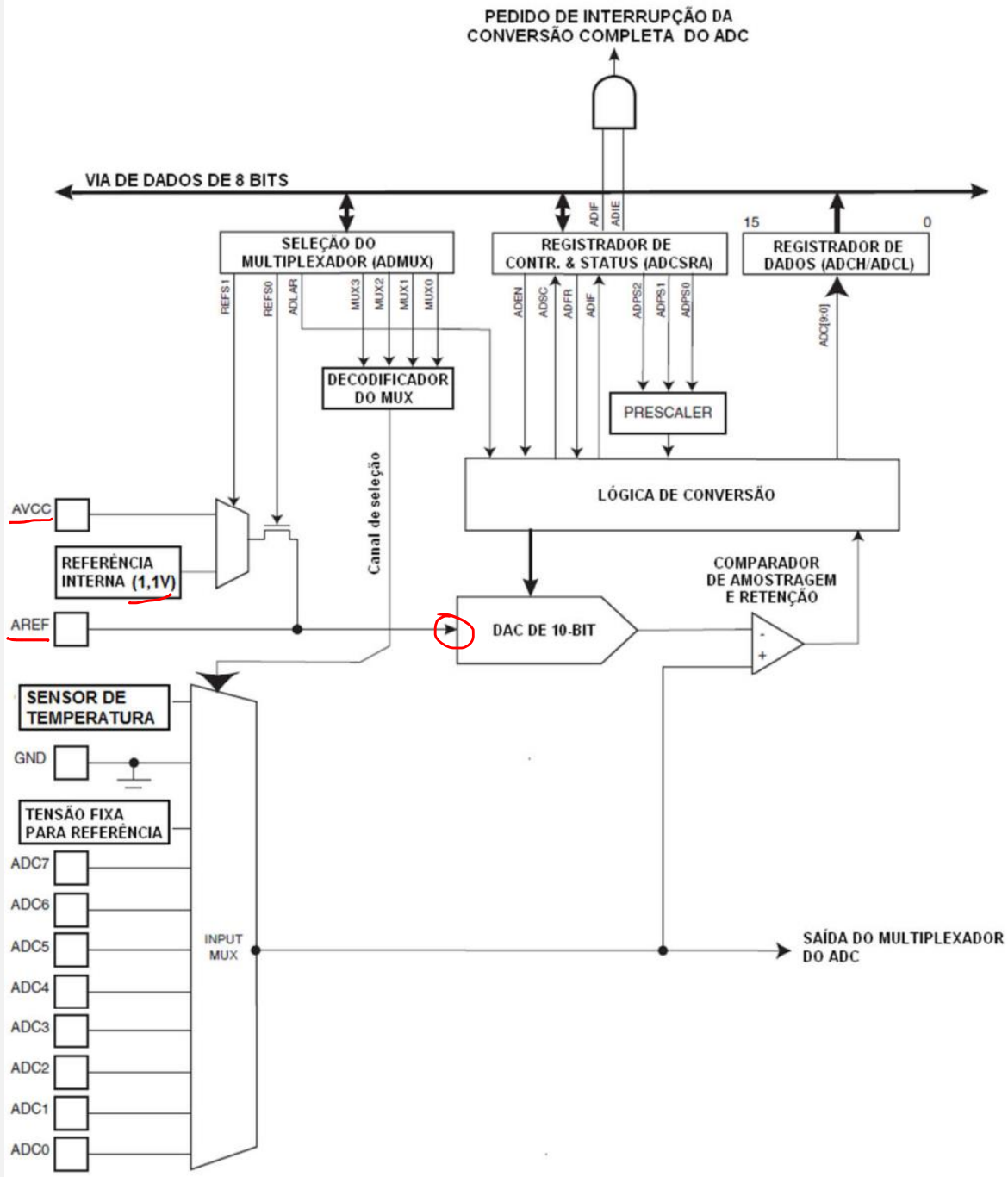


Resolução



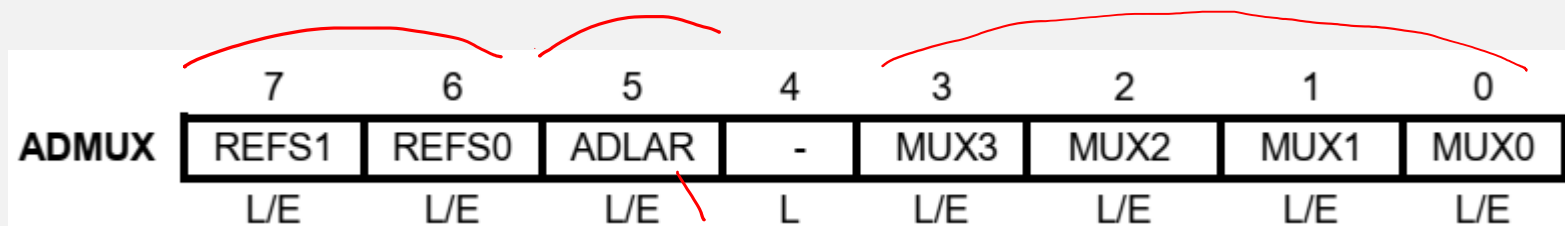
ADC NO ATMEGA328P

- 10 bits de resolução (1024 pontos).
- Precisão de ± 2 LSBs (bits menos significativos).
- Tempo de conversão de 65 até 260 μ s.
- Até 76,9 kSPS (kilo Samples Per Second), 15 kSPS na resolução máxima.
- 6 canais de entrada multiplexados (+2 nos encapsulamentos TQFP e QFN/MLF).
- Faixa de tensão de entrada de 0 até VCC.
- 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 °C de precisão.



REGISTRADORES DO ADC

ADMUX (ADC Multiplexer Selection Register)

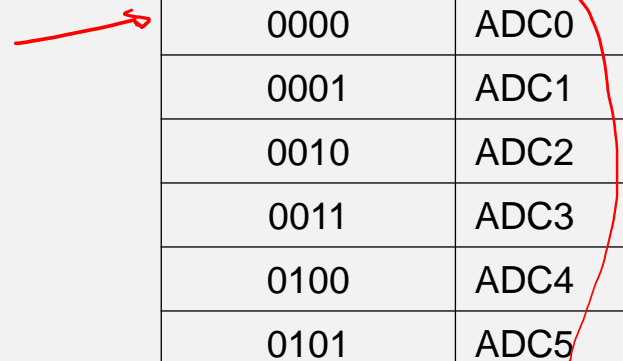


- Bits 7:6 – **REFS1:0** – Reference Selection Bit
- Bit 5 – **ADLAR** – ADC Left Adjust Result

REFS1	REFS0	Seleção da Tensão de Referência
0	0	AREF , tensão interna VREF desligada.
0	1	AVCC . Deve-se empregar um capacitor de 100 nF entre o pino AREF e o GND.
1	0	Reservado.
1	1	Tensão interna de referência de <u>1,1 V</u> . Deve-se empregar um capacitor de 100 nF entre o pino AREF e o GND.

REGISTRADORES DO ADC

- Bits 3:0 – **MUX3:0** – Analog Channel Selection Bits



MUX3..0	Entrada
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4
0101	ADC5
0110	ADC6
0111	ADC7
1000	Sensor interno de temperatura
1001-1101	reservado
1110	1,1 V (tensão fixa para referência)
1111	0 V (GND)

REGISTRADORES DA USART

ADCSRA (ADC Control and Status Register A)



- Bit 7 – ADEN – ADC Enable ✓
- Bit 6 – ADSC – ADC Start Conversion ✓
- Bit 5 – ADATE – ADC Auto Trigger Enable ✓
- Bit 4 – ADIF – ADC Interrupt Flag ✓
- Bit 3 – ADIE – ADC Interrupt Enable ✓
- Bits 2:0 – ADPS2:0 – ADC Prescaler Select Bits ✓

ADPS2	ADPS1	ADPS0	Fator de Divisão
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

Para (10bits) o taxa máx é 15kSPS
Cada amostra dura 13 clocks do ADC

Única opção em 16MHz
 $(16\text{MHz}/13)/128 = 9,6\text{kSPS}$
 $(16\text{MHz}/13)/64 = 19,2\text{kSPS}$

REGISTRADORES DA USART

ADCL/ADCH (ADC Data Register)

- Alinhamento a **Direita**

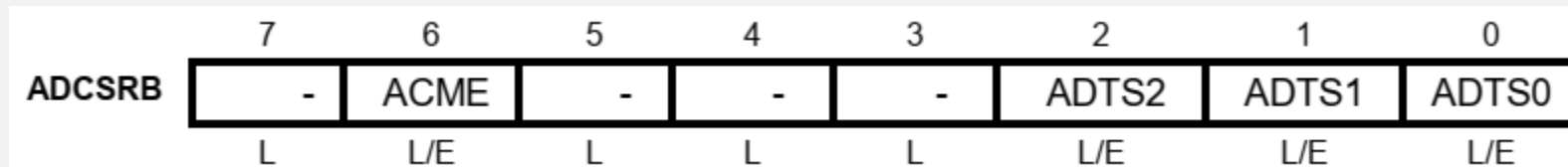
ADLAR=0	ADCH	15	14	13	12	11	10	9	8
		-	-	-	-	-	-	ADC9	ADC8
	ADCL	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0
Bit		7	6	5	4	3	2	1	0

- Alinhamento a **Esquerda**

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

REGISTRADORES DA USART

ADCSRB (ADC Control and Status Register B)



- Bits 2:0 – **ADTS2:0** - ADC Auto Trigger Source

ADTS2	ADTS1	ADTS0	Fonte de disparo
0	0	0	conversão contínua ✓
0	0	1	comparador Analógico ✓
0	1	0	interrupção Externa 0 ✓
0	1	1	igualdade de comparação A do TC0
1	0	0	estouro de contagem do TC0 ✓
1	0	1	igualdade de comparação B do TC1
1	1	0	estouro de contagem do TC1
1	1	1	evento de captura do TC1

REGISTRADORES DA USART

DIDR0 (Digital Input Disable Register 0)

	7	6	5	4	3	2	1	0
DIDR0	-	-	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D
	L/E	L/E	L/E	L/E	L/E	L/E	L/E	L/E

- Bits 5:0 – **ADC5D:0D** – ADC5:0 Digital Input Disable

```
#define F_CPU 16000000UL //Frequência de trabalho da CPU
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include "nokia5110.h"
```

```
#define tam_vetor 4
unsigned char leitura_ADC_string[tam_vetor];
uint16_t leitura_ADC = 0;
```

```
ISR(ADC_vect)
{
    leitura_ADC = ADC;
}

int main()
{
```

```
//GPIO
DDRB = 0xFF; //Porta B como saída
DDRC = 0x00; //Porta C como entrada
PORTC = 0xFE; //Desabilita o pullup do PC0
//Configura ADC
ADMUX = 0b11000000; //Tensão interna de ref (1.1V), canal 0
ADCSRA = 0b11101111; //habilita o AD, habilita interrupção,
modo de conversão contínua, prescaler = 128
ADCSRB = 0x00; //modo de conversão contínua
DIDR0 = 0b00111110; //habilita pino PC0 como entrada do ADC0
```

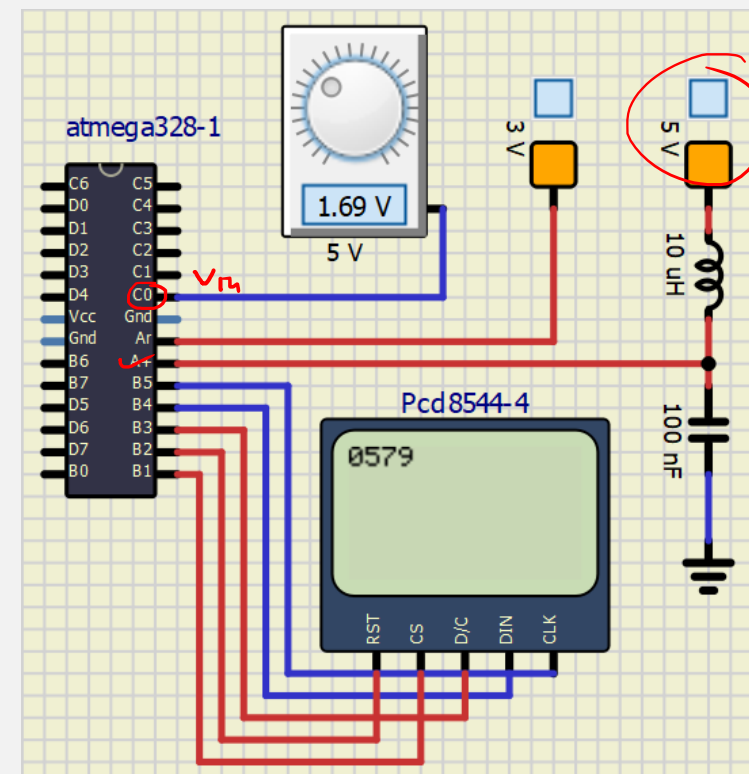
```
sei(); //Habilita interrupções globais
nokia_lcd_init(); //Inicia o LCD
```

```
while(1)
{
    nokia_lcd_clear(); //Limpa o LCD
    int2string(leitura_ADC, leitura_ADC_string);
    //converte a leitura do ADC em string
    nokia_lcd_write_string(leitura_ADC_string,1);
    //Escreve a leitura no buffer do LCD
    nokia_lcd_render(); //Atualiza a tela do display com
    o conteúdo do buffer
    _delay_ms(1000);
}
```

0	0	AREF
0	1	AVCC
1	1	Tensão interna de referência de 1,1 V.

```
//Conversão de inteiro para string
void int2string(unsigned int valor,
unsigned char *disp)
{
    for(uint8_t n=0; n<tam_vetor;
n++)
        disp[n] = 0 + 48;
    //limpa vetor para armazenagem
    dos dígitos
    disp += tam_vetor-1;
    do
    {
        *disp = (valor%10) + 48;
        //pega o resto da
        divisão por 10
        valor /=10; //pega o
        inteiro da divisão por
        10
        disp--;
    }while (valor!=0);
}
```

EXEMPLO: ADC



REFERÊNCIAS

IDE

- Atmel Studio 7 (gratuito) <https://www.microchip.com/mplab/avr-support/atmel-studio-7>

Simuladores

- <https://www.simulide.com/p/blog-page.html>
- <https://github.com/lcgamboa/picsimlab/releases>
- <https://www.labcenter.com/downloads/>

Material de referência:

- Datasheet do Atmega 328p: <https://www.microchip.com/wwwproducts/en/ATmega328p#datasheet-toggle>
- Livro texto: <http://borgescorporation.blogspot.com/2012/05/avr-e-arduino-tecnicas-de-projeto.html>