

Conversor Analógico/Digital (A/D ou ADC)

DCA0119 - Sistemas Digitais

Mário Sérgio Cavalcante¹

¹ Departamento de Engenharia de Computação e Automação
mariocavalcante@dca.ufrn.br

Natal - Rio Grande do Norte

Sumário

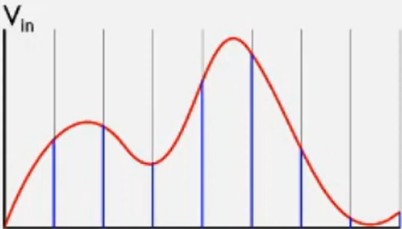
- 1 Introdução
- 2 Registradores ADCH e ADCL
- 3 Registrador ADCSRA
- 4 Registrador ADMUX
- 5 Exemplo
 - Definições Gerais
 - Exemplo 1
 - Exemplo 2

Introdução

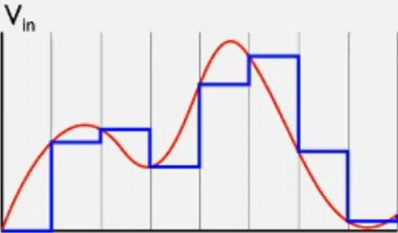
Conversor A/D



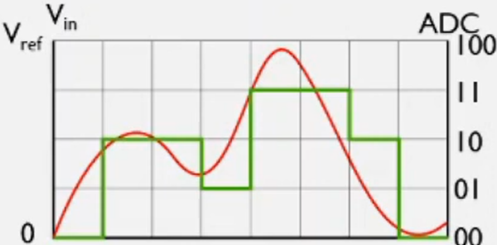
Sinal Analógico



Sampling



Sampling and Hold

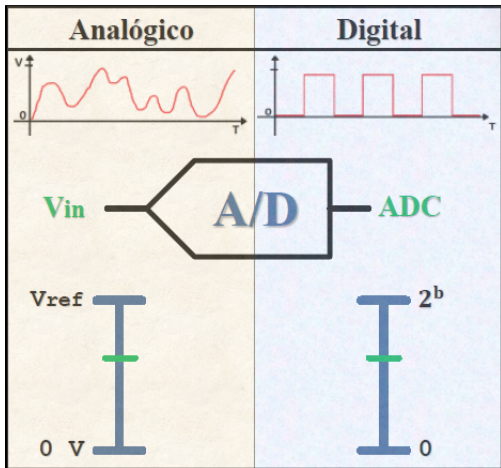


Quantization

Introdução

- Necessário para processamento de sinais externos de grandezas analógicas por sistemas digitais.
- O Atmega328p incorpora um conversor A/D, com as seguintes características:
 - 10 bits de resolução (1024 pontos);
 - Tempo de conversão de 13 até 260 μ s;
 - Até 76,9 kSPS (kilo amostras por segundo);
 - 6 canais de entrada multiplexados;
 - Tensão de referência selecionável;
 - Faixa de tensão de entrada de 0 até V_{cc} ;
 - Modo de conversão simples ou contínua;
 - Interrupção ao término da conversão;
 - Eliminador de ruído para o modo Sleep.

Conversor A/D - Conversão

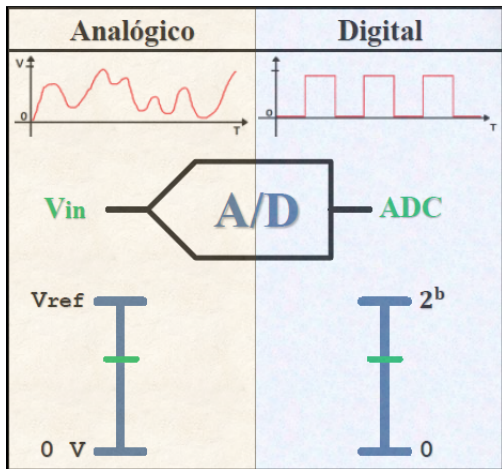


V_{in} - Tensão de Entrada;
 V_{ref} - Tensão de referência;
 ADC - Saída do conversor

b - É o número de bits do A/D;
 Para $b = 10$, temos:
 então $2^{10} = 1024$

Degrau de Resolução:
 $1\text{ LSB} = \frac{V_{ref}}{2^b}$

Conversor A/D - Conversão



$$ADC = \frac{V_{\text{in}} \cdot 2^b}{V_{\text{ref}}}$$

$$ADC = \frac{V_{\text{in}} \cdot 1024}{V_{\text{ref}}}$$

Conversor A/D

Observação 1.1

- A maior tensão de entrada V_{in} deve ser V_{ref} menos 1 LSB:
- Para um A/D de 10 bits e $V_{ref} = 5\text{ V}$:
- Maior valor da saída A/D é de 1023.
- Degrau de resolução:

$$1\text{ LSB} = 5/1024 = 4,88\text{ mV}$$

- Maior valor de v_{in} deve ser 4,991 \rightarrow 1023



$$ADC = \frac{V_{in} \cdot 1024}{V_{ref}}$$

Conversor A/D - Conversão

1.1: Conversão

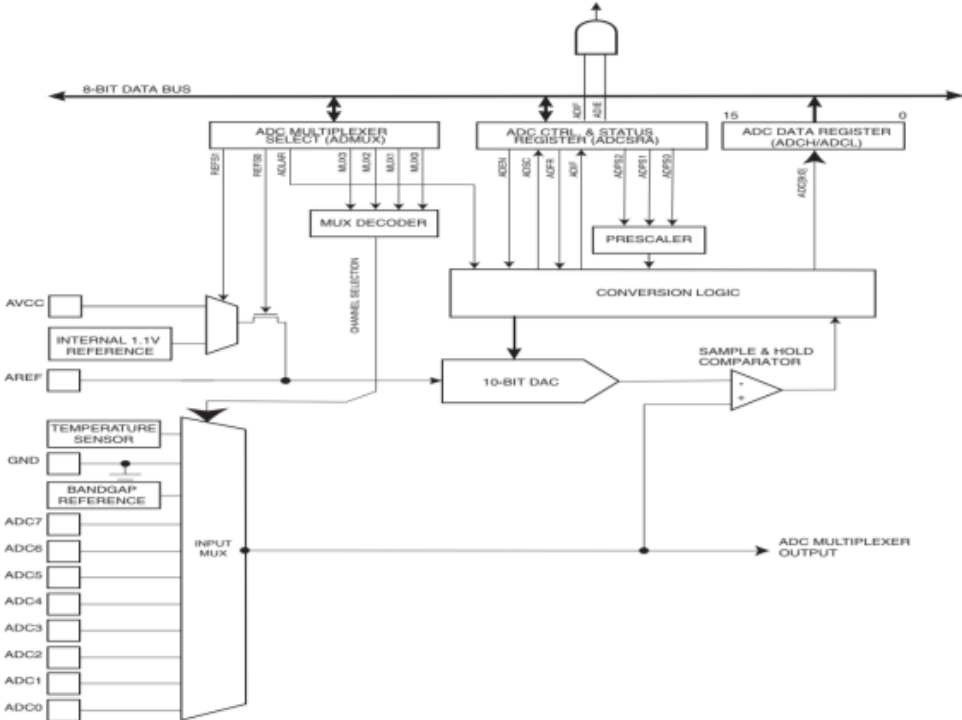
Em um Atmega328p, se for empregada a tensão interna de referência de 1,1 V e uma tensão de entrada for de 168 mV, qual o valor correspondente convertido pelo A/D? Qual o degrau de resolução do A/D?

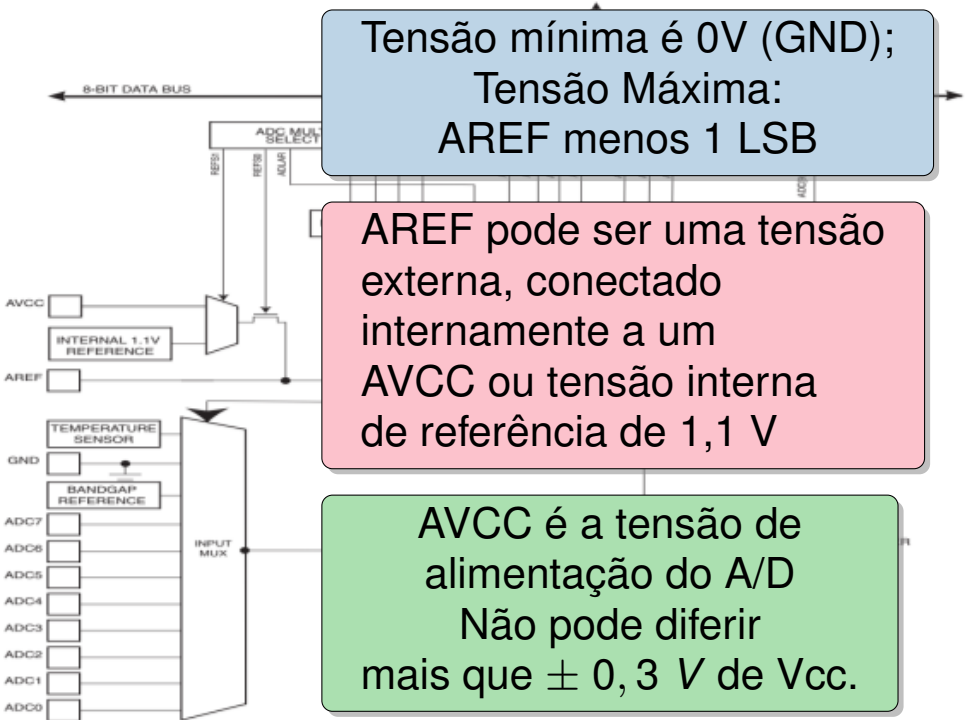
$$ADC = \frac{0,168 \cdot 1024}{1,1} = 156 = 0b10011100$$

$$1 \text{ LSB} = \frac{V_{ref}}{1024} = \frac{1,1}{1024} = 1,07 \text{ mV}$$



$$ADC = \frac{V_{in} \cdot 1024}{V_{ref}}$$

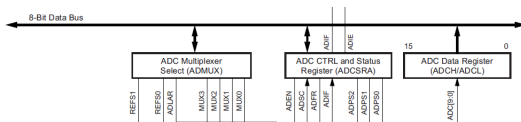




Registradores ADCH e ADCL

2.1: Resultado

- Os registradores ADCH e ADCL recebem resultado da conversão
- Bit ADLAR do registrador ADMUX configura a forma de apresentação do resultado.
- Quando ADCL é lido, ADCL e ADCH são bloqueados até a leitura de ADCH.
- Conversão iniciada quando ADSC do registrador ADCSRA é colocado em 1 Colocado automaticamente em 0 no final da conversão.



Justificado à direita (padrão).

ADCH								ADCL							
-	-	-	-	-	-	9	8	7	6	5	4	3	2	1	0

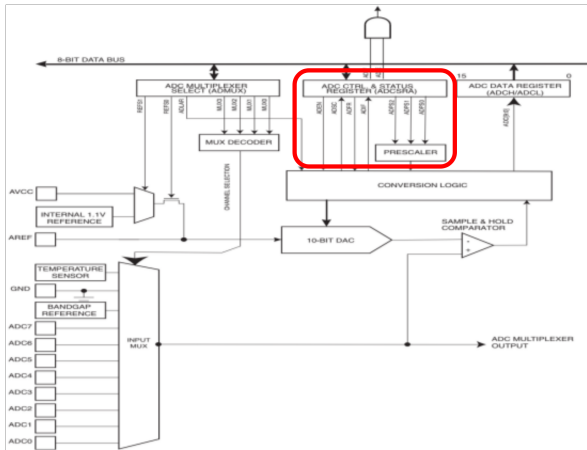
Justificado à esquerda.

ADCH								ADCL							
9	8	7	6	5	4	3	2	1	0						

Registrador ADCSRA

3.1: Clock

- Para o circuito de conversão sucessivas do A/D obter a máxima resolução, é requerida uma frequência de 50 a 200 kHz;
- A partir do clock da CPU, um prescaler produz um clock para o A/D, que é configurado pelos bits ADPS em ADCSRA;
- O prescaler funciona a partir do momento em que o A/D é ligado, colocando em 1 o bit ADEN em ADCSRA



Registrador ADCSRA

Bit	7	6	5	4	3	2	1	0
ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
Lê/Escreve	L/E	L/E	L/E	L/E	L/E	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

- **Bit 2:0 - ADC Prescaler Select Bits** - Determinam a divisão do *clock* da CPU para o *clock*, conforme a tabela:

ADPS2	ADPS1	ADPS0	Fator de Divisão
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8

ADPS2	ADPS1	ADPS0	Fator de Divisão
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Registrador ADCSRA

Bit	7	6	5	4	3	2	1	0
ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
Lê/Escreve	L/E	L/E	L/E	L/E	L/E	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

- **Bit 3 – ADIE – ADC Interrupt Enable** – Este bit habilita a interrupção do A/D após a conversão se o bit I do SREF estiver habilitado.

Registrador ADCSRA

Bit	7	6	5	4	3	2	1	0
ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
Lê/Escreve	L/E	L/E	L/E	L/E	L/E	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

- **Bit 4 – ADIF – ADC Interrupt Flag** – Este bit é ativo quando uma conversão for completada e o registrador de dados atualizado.

Registrador ADCSRA

Bit	7	6	5	4	3	2	1	0
ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
Lê/Escreve	L/E	L/E	L/E	L/E	L/E	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

- **Bit 5 – ADATE** – **ADC Auto Trigger Enable** – Ativa o modo de auto disparo. O A/D começará uma conversão na borda positiva do sinal selecionado de disparo. A fonte de disparo é selecionada nos bits **ADTS2:0** do registrador ADCSRB.

Registrador ADCSRA

Bit	7	6	5	4	3	2	1	0
ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
Lê/Escreve	L/E	L/E	L/E	L/E	L/E	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

- **Bit 6 – ADSC – ADC Start Conversion** – No modo de conversão simples, ADSC = 1 irá iniciar uma conversão. No modo de conversão contínuo, ADSC = 1 irá iniciar a primeira conversão. ADSC ficará em 1 durante todo o processo de conversão e será zerado automaticamente ao seu término. Se o ADSC é escrito ao mesmo tempo em que o A/D é habilitado, a primeira conversão levará 25 ciclos de clock do A/D ao invés de 13 normais.

Registrador ADCSRA

Bit	7	6	5	4	3	2	1	0
ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
Lê/Escreve	L/E	L/E	L/E	L/E	L/E	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

- **Bit 7 – ADEN – ADC Enable** – Habilita o A/D. Se ADEN = 0, o A/D é desligado. Desligar o A/D durante uma conversão irá finalizá-la.

ADCSRB

Bit	7	6	5	4	3	2	1	0
ADCSRB	-	ACME	-	-	-	ADTS2	ADTS1	ADTS0
Lê/Escreve	L	L/E	L	L	L	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

- **Bit 2:0 – ADTS2:0 – ADC Auto Trigger Source** – Se o bit ADATE no registrador ADCSRA for 1, o valor desses bits seleciona a fonte para o disparo da conversão. Se ADATE é zero esses bits não têm efeito.

ADCSRB

Bit	7	6	5	4	3	2	1	0
ADCSRB	-	ACME	-	-	-	ADTS2	ADTS1	ADTS0
Lê/Escreve	L	L/E	L	L	L	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

Bit 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

Registrador DIDR0

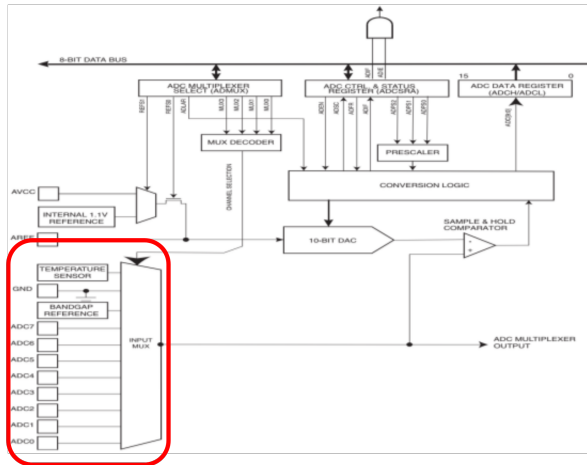
Bit	7	6	5	4	3	2	1	0
DIDR0	-	-	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D
Lê/Escreve	L/E	L/E	L/E	L/E	L/E	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

- **Bit 2:0 – ADTS2:0 – Bit 5:0 – ADC5D:0D – ADC5:0 Digital Input Disable** – Estes bits desabilitam individualmente as entradas digitais dos pinos do A/D. Deve(m) estar ativo(s) sempre que o pino correspondente for utilizado como entrada para o A/D, caso o contrário deve(m) estar em zero.

Registrador ADMUX

4.1: Mux

- O registrador ADMUX, que é responsável pela mudança do canal de entrada do conversor pode ser atualizado seguramente quando:
- O bit ADSC ou o bit ADEN for igual a zero;
- Durante a conversão, pelos menos um ciclo de clock do A/D após o início da conversão;
- Após a conversão, antes do flag de interrupção utilizado como disparo ser limpo.



ADMUX - Registrador

Bit	7	6	5	4	3	2	1	0
ADMUX	REFS1	REFS0	ADLAR	-	MUX3	MUX2	MUX1	MUX0
Lê/Escreve	L/E	L/E	L/E	L/E	L/E	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

- **Bit 3:0 – MUX3:0 – Analog Channel Selection Bits** - Seleccionam qual entrada analógica será conectada ao A/D.

MUX3..0	Entrada
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4
0101	ADC5

MUX3..0	Entrada
0101	ADC5
0110	ADC6
0111	ADC7
1000	Sensor Interno de Temperatura
1001-1010	Reservado
1110	1,1 (Tensão Fixa para referência)
1111	0V (GND)

Registrador ADMUX - ADLAR

Bit	7	6	5	4	3	2	1	0
ADMUX	REFS1	REFS0	ADLAR	-	MUX3	MUX2	MUX1	MUX0
Lê/Escreve	L/E	L/E	L/E	L/E	L/E	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

- **Bit 5 – ADLAR – ADC Left Adjust Result** - Afeta a representação do resultado da conversão dos registradores de dados do A/D. ADLAR = 1, alinhado à esquerda; ADLAR = 0, alinhado à direita. A alteração deste bit afeta imediatamente os registradores de dados.

Registrador ADMUX - REFS1:0

Bit	7	6	5	4	3	2	1	0
ADMUX	REFS1	REFS0	ADLAR	-	MUX3	MUX2	MUX1	MUX0
Lê/Escreve	L/E	L/E	L/E	L/E	L/E	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

- **Bit 7:6 – REFS1:0 – Reference Selection Bit** - Estes bits selecionam a fonte de tensão para o A/D, conforme a tabela. Se uma mudança ocorrer durante uma conversão, a mesma não terá efeito até a conversão ser completada (ADIF no ADCSRA estar ativo).

4.2: Atenção

A referência interna não pode ser utilizada se uma tensão externa estiver sendo aplicada ao pino AREF.

Registrador ADMUX - REFS1:0

Bit	7	6	5	4	3	2	1	0
ADMUX	REFS1	REFS0	ADLAR	-	MUX3	MUX2	MUX1	MUX0
Lê/Escreve	L/E	L/E	L/E	L/E	L/E	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

■ Bit 7:6 – REFS1:0 – Reference Selection Bit

REFS1	REFS0	Descrição
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 1,1 V. Deve-se empregar um capacitor de 100 nF entre o pino AREF e o GND.

Exemplo

Configurações Iniciais

```

4 //-----
5 //Configura o conversor ADC do ATmega328p
6 //ref = 0. Para usar a tensão de referência Aref
7 //ref = 1. Para usar a tensão de referência Avcc - Lembre-se do Capacitor 100nF
8 //ref = 2. Para usar a tensão de referência interna de 1,1 V
9 //did: valor para o registrador DIDR0
10 //-----
11 void adcBegin(uint8_t ref, uint8_t did)
12 {
13     ADCSRA = 0; //configuração inicial
14     ADCSRB = 0; //configuração inicial
15     DIDR0 = did; //configuração DIDR0
16     if (ref == 0)
17     {
18         ADMUX &= ~(1<<REFS1) | (1<<REFS0); //Aref
19     }
20     if ((ref == 1) || (ref > 2))
21     {
22         ADMUX &= ~(1<<REFS1); //Avcc
23         ADMUX |= (1<<REFS0); //Avcc
24     }
25     if (ref == 2)
26     {
27         ADMUX |= (1<<REFS1) | (1<<REFS0); //Tensão interna de ref (1.1V)
28     }
29     ADMUX &= ~(1<<ADLAR); //Alinhamento a direita
30     ADCSRA |= (1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0); //habilita AD. Prescaler de 128 (clk_AD = F_CPU/128)
31 }

```

Selecionando o canal

```

29 //-----|
30 //Seleciona canal do ADC
31 //0 <= channel <= 5 - Leitura dos pinos AD0 a AD5
32 //channel = 6 - Leitura do sensor de temperatura
33 //channel = 7 - 1,1V
34 //channel > 7 - GND
35 //-----
36 void adcChannel(uint8_t canal)
37 { if (canal <= 5)//seleciona um canal no multiplex
38     ADMUX = (ADMUX & 0xF0) | canal;
39     if (canal == 6)//seleciona sensor interno de temperatura
40     ADMUX = (ADMUX & 0xF0) | 0x08;
41     if (canal == 7)//seleciona 1,1 V
42     ADMUX = (ADMUX & 0xF0) | 0x0E;
43     if (canal > 7)//seleciona GND
44     ADMUX = (ADMUX & 0xF0) | 0x0F;
45 }
```

Inicia o processo de conversão

```
47 //-----
48 //Inicia conversão
49 //-----
50 void adcSample(void)
51 { ADCSRA |= (1<<ADSC); //Inicia conversão
52 }
53
54 //-----
55 //Verifica se conversão foi concluída
56 //Retorna valor 0 se conversão concluída. 64 se não.
57 //-----
58 uint8_t adcOk(void)
59 { return (ADCSRA & (1<<ADSC));
60 }
```

Ler o valor do ADC

```

73 //-----
74 //Ler o ADC e retorna o valor lido do ADC
75 //-----
76 uint16_t adcReadOnly()
77 { return ((ADCH<<8) | ADCL); //retorna o valor do ADC
78 }
79
80 //-----
81 //Converte, aguarda, ler e retorna valor lido do ADC
82 //-----
83 uint16_t adcRead()
84 { adcSample(); //Inicia conversão
85   while(adcOk()); //Aguarda fim da conversão (ADSC = 0)
86   return adcReadOnly(); //retorna o valor do ADC
87 }

```

Habilita a interrupção

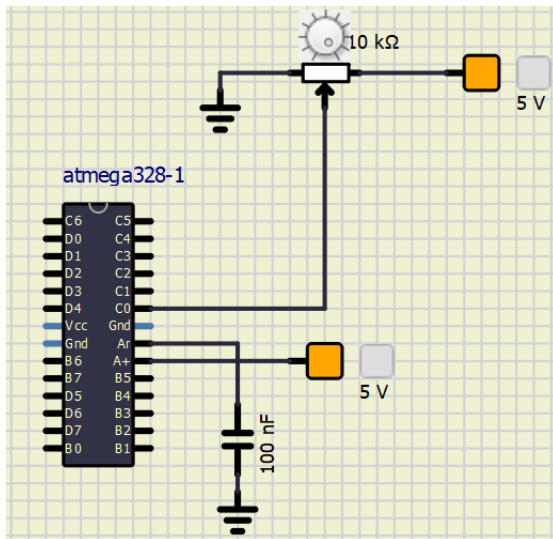
```
77 //-----
78 //Habilita ou desabilita interrupção do ADC
79 //Se x = 0, desabilita interrupção
80 //Caso contrário, habilita interrupção
81 //-----
82 void adcIntEn(uint8_t x)
83 { if (x)
84     ADCSRA |= (1<<ADIE); //habilita interrupção do ADC
85     else
86     ADCSRA &= ~(1<<ADIE); //Desabilita interrupção do ADC
87 }
```

Define as entradas

```

2  #define AREF 0 // Tensão de referência = Aref
3  #define AVCC 1 // Tensão de referência = AVcc
4  #define VR11 2 // Tensão de referência = 1,1 V
5  #define ADC0 0 // Seleciona a entrada ADC0
6  #define ADC1 1 // Seleciona a entrada ADC1
7  #define ADC2 2 // Seleciona a entrada ADC2
8  #define ADC3 3 // Seleciona a entrada ADC3
9  #define ADC4 4 // Seleciona a entrada ADC4
10 #define ADC5 5 // Seleciona a entrada ADC5
11 #define Vtemp 6 // Seleciona o Sensor de Temperatura
12 #define V11 7 // Seleciona a tensão de 1,1 V
13 #define Vgnd 8 // Seleciona a tensão GND (0 V)
    
```

Circuito - Exemplos



Exemplo 1: Código

```
278 int main(void)
279 { uint16_t valorADC;
280   UART_Init();//Inicialização UART
281   adcBegin(AVCC, 0x01); //Inicialização A/D
282   adcChannel(ADC0);      //seleciona entrada
283   while (1)              //laço infinito
284   { valorADC = adcRead();//lê o valor analógico
285     uartDec2B(valorADC); //envia valor lido
286     uartString("\r\n"); //nova linha
287     _delay_ms(200);      //aguarda
288   }
289 }
```


Exemplo 2: Código

```
int main(void)
{
    UART_Init(); //Inicializa UART
    adcBegin(AVCC, 0x01); //Inicializa A/D
    adcChannel(ADC0); //Seleciona canal
    adcIntEn(1); //Interrupção do A/D
    sei(); //Interrupção geral
    while (1) //Laço infinito
    {
        if (uartRxOk()) //verifica se existe novo
            switch(uartRx()){
                case 'C':
                case 'c':
                    adcSample(); //Inicia conversão
            }
    }
}

//Tratamento da interrupção do A/D
ISR(ADC_vect)
{
    uartString("Valor: "); //Envia string
    uartDec2B(adcReadOnly()); //Ler e envia valor do A/D
    uartString("\r\n"); //Nova linha
}
```

Neste exemplo, a conversão da tensão aplicada ao pino ADC0 (PC0) é iniciada quando recebe a letra "C" ou a letra "c" pela porta UART. A leitura do valor convertido é lido após interrupção de fim da conversão.

Exemplo 3: Código

```

280 int main(void)
281 {
282     UART_Init(); //Inicializa UART
283     adcBegin(AVCC, 0x01); //Inicialização A/D
284     adcChannel(ADC0); //Seleciona canal
285     adcIntEn(1); //Interrupção do A/D
286     uartIntRx(1); //Interrupção da uart
287     sei(); //Interrupção geral
288     while (1);
289 }
290 //Tratamento da interrupção de recepção de dados
291 ISR(USART_RX_vect)
292 {
293     switch (uartRx()){
294     case 'c':
295     case 'C':
296         adcSample(); //Inicia conversão
297         break;}
298 }
299 //Tratamento da interrupção do A/D
300 ISR(ADC_vect)
301 {
302     uartString("Valor: "); //Envia string
303     uartDec2B(adcReadOnly()); //Ler e envia valor do A/D
304     uartString("\r\n"); //Nova linha
305 }

```

Neste exemplo, duas interrupções são utilizadas. Uma após a recepção de dados pela UART e outra após a conversão do A/D for finalizada. O funcionamento é semelhante ao exemplo anterior, mas a função principal fica "livre" para realizar outras atividades até a recepção de um dado pela UART.