

# Voltímetro com a porta paralela

Alexandre Erwin Ittner  
kernel32@bigfoot.com  
<http://kernel32.cjb.net>  
Outubro / 2000

## 1. Introdução

Este circuito simples faz a leitura de um sinal de tensão através de um conversor analógico-digital conectado à porta paralela de um PC.

O software escrito em Pascal roda em ambiente DOS e monitora constantemente a evolução do sinal de tensão externo. A comunicação entre o conversor analógico-digital e a porta paralela é feita através de um multiplexador.

## 2. Software

```
{
  voltimetro - (c) Alexndre Erwin Ittner - 2000
  kernel32@bigfoot.com - http://kernel32.cjb.net

  Este programa e' distribuido nos termos da Licenca
  Publica Geral GNU (GPL) versao 2. SEM NENHUMA
  GARANTIA!
}

program voltimetro;
uses crt;

const tempo_espera = 1; { em ms }
      atenuacao = 5;     { Relacao de atenuacao entre o resistor
                        multiplicador e a impedancia de entrada
                        do conversor }

      porta_dados = $378; { Porta de escrita }
      porta_status = $379; { Porta de leitura }

var posicao_bit:integer;
    leitura_bits:integer;

{ Funcoes de escrita e leitura a porta com assembler }
procedure escreve_porta(porta:integer;dados:integer); far;
begin
  asm
    mov ax,dados
    mov dx,porta
    out dx,ax
  end;
end;
```

```

function le_porta(porta:integer):integer; far;
var  dadoslidos:integer;
begin
    asm
        mov dx,porta
        in ax,dx
        mov dadoslidos,ax
    end;
le_porta := byte(dadoslidos);
end;

{ jah que o pascal nao tem potencia, vamos implementar }
function potencia(x:real; y:integer):integer;
begin
potencia := trunc(exp(ln(abs(x)) * y));
end;

begin
    while not keypressed do
        begin
            leitura_bits := 0;

            { Envia sinal ao pino CONVERT do conversor. O sinal de saida
              ficara travado na entrada do MUX, podendo ser acessado
              bit a bit }
            escreve_porta(porta_dados, 8);

            delay(tempo_espera);

            { Le bit a bit o sinal do conversor, enderecando cada
              bit atraves do multiplexador }

            for posicao_bit := 0 to 7 do
                begin
                    escreve_porta(porta_dados, posicao_bit + 128);
                    delay(tempo_espera);

                    { faz um AND logico para verificar se o 8o. bit da porta
                      de status esta setado. Este bita JA ESTA INVERTIDO
                      PELO OPTOACOPLADOR }
                    if (le_porta(porta_status) and 128 ) = 128 then
                        leitura_bits := leitura_bits +
                            potencia(2, posicao_bit);

                    { por seguranca, espera antes de ler o proximo }
                    delay(tempo_espera);
                end;

            escreve_porta(porta_dados, 0); { limpa a saida }

            clrscr;
            writeln(' Voltmetro - (c) Alexandre Erwin Ittner - 2000');
            writeln(' Distribuido nos termos da GPL 2 - SEM GARANTIAS');
            writeln('');

            { exibe a leitura de tensao formatada, com
              as casas decimais fixas }
            writeln(' Tensao: ', trunc(leitura_bits * atenuacao *

```

```

        (10/255)), ',', trunc(100 * frac(leitura_bits *
        atenuacao * (10/255))), 'V');

    if leitura_bits >= 255 then
        writeln('          SOBRETENSAO!')
    else
        writeln('');

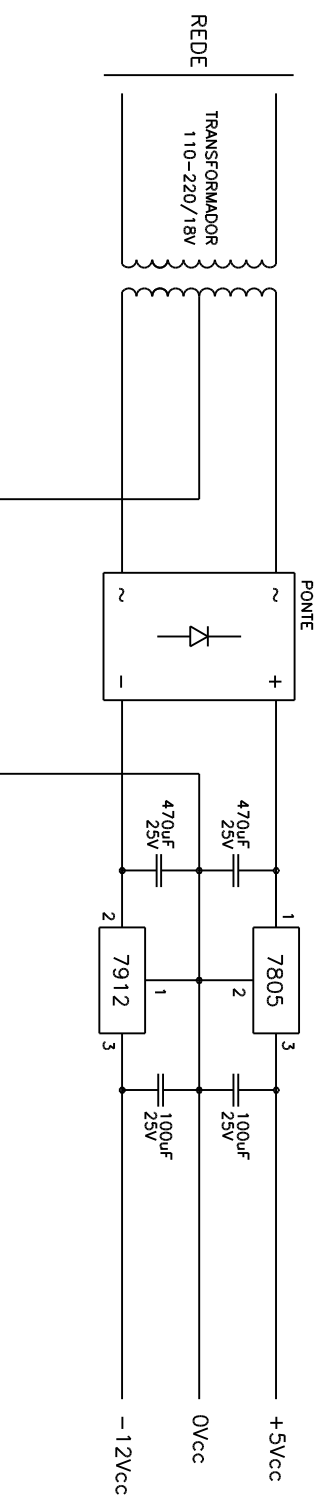
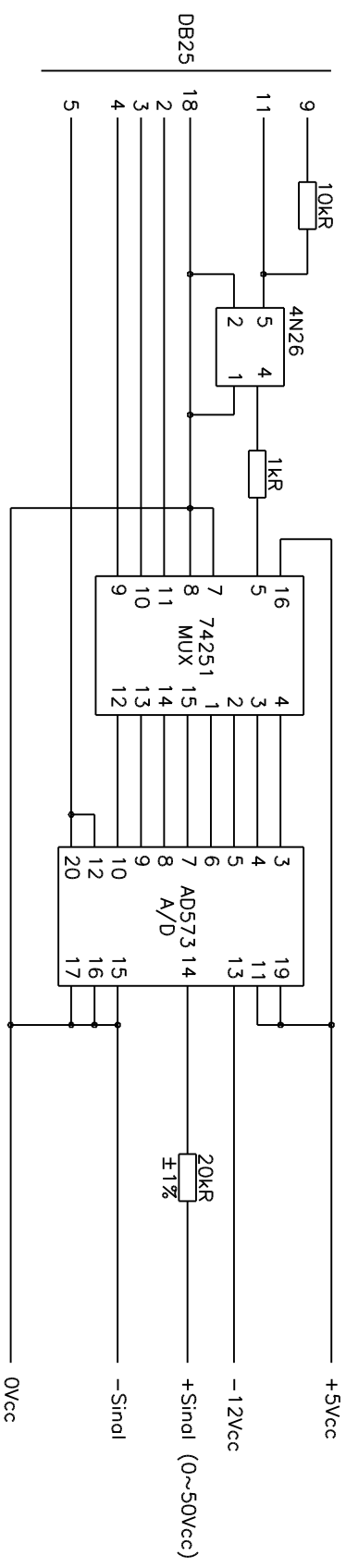
        writeln(' Maxima tensao de entrada: ', 10 * atenuacao, 'Vcc');
        writeln(' Atenuacao do sinal de entrada: ', atenuacao, 'X');
        writeln(' Erro do conversor: ',
            trunc(1000 * atenuacao * (10/255)), 'mV');

        delay(100 * tempo_espera);
    end;

    { Limpa a saida antes de finalizar }
    escreve_porta(porta_dados, 0);
    clrscr;
end.

```

### 3. Hardware



VOLTIMETRO COM A PORTA PARALELA

Alexandre Erwin Iltnner – 2000  
 kernel32@bigfoot.com  
 http://kernel32.cjb.net