

Simple osciloscópio para sinais digitais

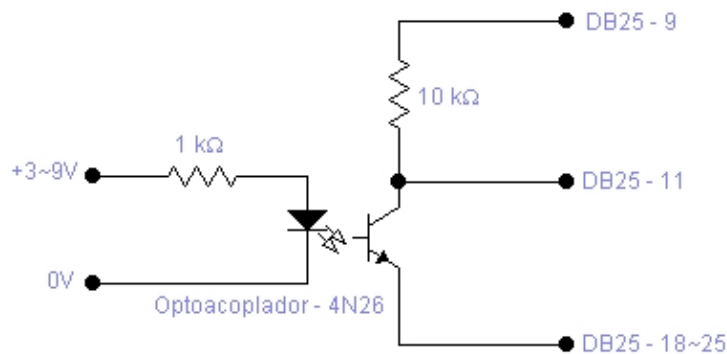
Alexandre Erwin Ittner – <http://kernel32.cjb.net>

Notas importantes

Não me responsabilizo pela montagem e funcionamento deste circuito. Também não garanto que o circuito que você montar irá funcionar. Testei esse circuito exatamente da forma como está descrito sem nenhum problema, mas não posso me responsabilizar pelos seus atos

Descrição de funcionamento

Para tornar possível a monitoração via porta paralela de um sinal digital em função do tempo, imaginei um circuito que envie um sinal isolado ao bit BUSY (pino 11) da porta paralela. Um software rodando em ambiente gráfico se encarregaria de monitorar constantemente este bit e traçar a evolução do sinal ao longo do tempo. A melhor conclusão que cheguei foi:



Neste circuito existe uma isolação total entre o sinal externo e o PC. O sinal responsável por acionar o bit BUSY provém do 7o. bit de dados (pino 9), evitando a necessidade de uma fonte externa (sempre perigosa em montagens deste tipo).

A conexão do circuito à porta paralela deverá ser feita com um conector DB25, as conexões estão indicadas ao lado dos terminais do optocouplador.

Existem limitações de tensão na entrada do circuito, que se desobedecidas levarão à destruição do optocouplador. De forma geral, **jamais insira um sinal superior a 9V na entrada**. Sinais inferiores a 3V poderão não acionar corretamente o optocouplador.

O software de monitoração se encarregará de setar o 7o. bit de dados, antes de efetuar a leitura do bit BUSY

O conector DB25 possui a seguinte pinagem



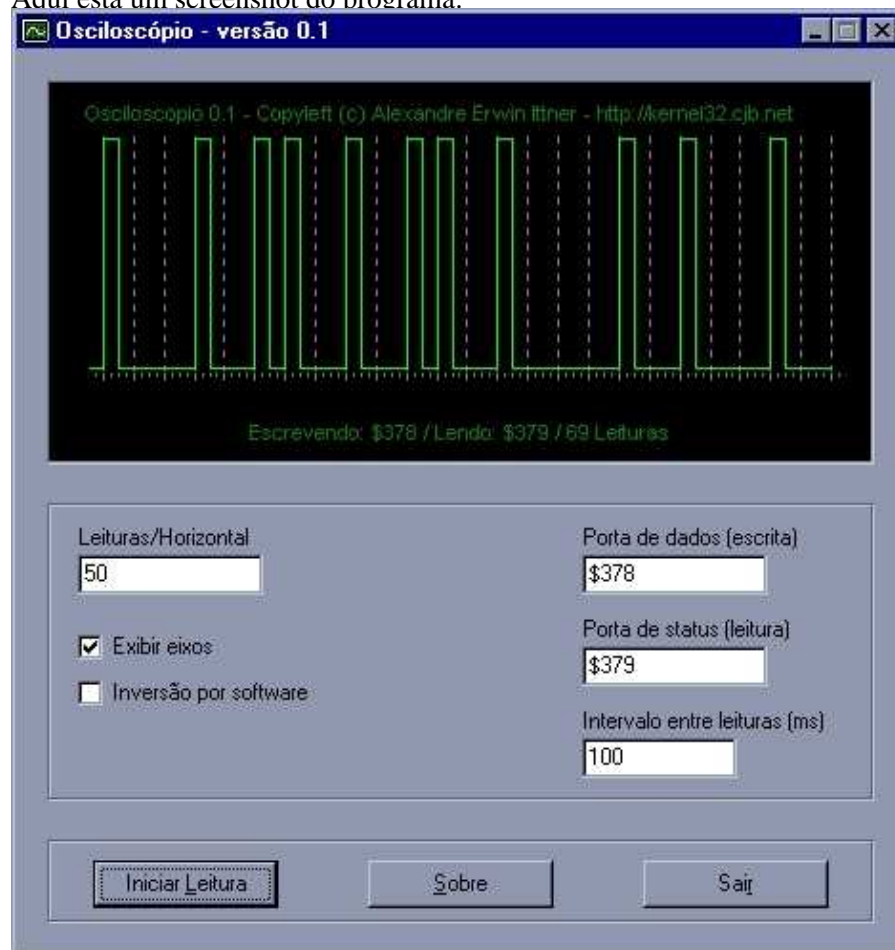
A montagem do circuito deve ser feita em placa de circuito impresso ou em placa-protótipo. No meu caso, fiz uma placa suficientemente pequena para ser colocada dentro do conector DB25, dando um excelente resultado funcional e estético.

Software

O software que escrevi para monitorar o sinal de entrada está atualmente disponível para a plataforma Windows :-(

Este foi escrito em linguagem Delphi (clássica pelo suporte a banco de dados e muito usada em aplicações comerciais). Não tenho conhecimento em C++, Tcl/tk, QT, ou GTK, portanto uma versão para a plataforma UNIX não estará disponível.

Aqui está um screenshot do programa:



O software, bastante simples, permite alguns ajustes, a seguir:

Porta de dados: Ajuste da porta de dados, usualmente 378h (888d). A entrada de um valor hexadecimal deve

ser feita ao estilo Pascal (\$valor).

Porta de status: Ajuste da porta de status, responsável pela aquisição do sinal enviado pela porta de dados. Usualmente 379h (889d). A entrada de um valor hexadecimal deve ser feita ao estilo Pascal (\$valor).

Intervalo entre leituras: O período, em milissegundos, que o programa deverá esperar entre o fim de uma leitura e o início da próxima.

Leituras/Horizontal: O número de leituras que serão exibidas no painel de visualização

Exibir eixos: Ativa ou desativa a exibição dos eixos no painel de visualização

Inversão por software: Indica se o sinal de entrada deverá ser exibido de forma invertida.

Direitos Autorais

Este circuito não é nenhuma concepção nova e é largamente utilizado, não existe nenhuma restrição à sua distribuição. Evidentemente, se algum projeto for derivado diretamente deste a fonte deverá ser citada.

O software idealizado é distribuído de acordo com a General Public License (GPL), versão 2 ou superior, conforme especificado pela [Free Software Foundation](http://www.gnu.org/licenses/gpl-2.0.html).

De forma simplificada:

Este programa é um software de livre distribuição, que pode ser copiado e distribuído sob os termos da Licença Pública Geral GNU (GNU General Public License – GPL) versão 2, conforme publicada pela Free Software Foundation. Este programa é distribuído na expectativa de ser útil aos seus usuários, porém NÃO HÁ NENHUMA GARANTIA, EXPLÍCITA OU IMPLÍCITA, COMERCIAL OU DE ATENDIMENTO A UMA DETERMINADA FINALIDADE. Consulte a Licença Pública Geral GNU (GNU General Public License – GPL) para maiores detalhes.

#EOF