

EA201 - EXERCÍCIO PRÁTICO 3

INTRODUÇÃO:

Neste experimento, vamos analisar os sinais gerados por uma MCU na sua interface SPI, para configurar e utilizar um *display* LCD monocromático gráfico de 84x48 pontos. Como nos outros experimentos, é necessário baixar o arquivo contendo a captura de dados e abri-lo no Logic. Também será necessário baixar o *datasheet* do controlador do módulo do *display* (Um microcontrolador dedicado da Philips ® denominado PCD8544). Este módulo é conhecido como “Nokia 5110”, pois era usado neste modelo de telefone celular, bem como outros da mesma empresa. Ele possui uma interface SPI unidirecional (não há dados transmitidos pelo módulo para a MCU), atuando exclusivamente como *slave*. Esta interface é composta dos pinos CLK (*clock*), DI (entrada de dados, ligada ao sinal MISO) e CE (*Chip Enable*). Além destes sinais da interface SPI, há ainda um RST (Reset geral) e um DC (*Data / Command*), que define se o byte enviado pelo *Master* é um comando / configuração (DC = *Low*) ou se é um dado que determina pixels a serem ativados no *display* (DC = *High*).

No *datasheet*, o importante é se concentrar nas páginas 6 a 16 para entender o funcionamento do *display*. Note que para escrever texto, é necessário “desenhar” as letras. Para isto ser feito com uma fonte básica de caracteres 7x5, normalmente se ajusta o modo de endereçamento horizontal e enviando cinco colunas de 8 pixels (com o bit mais significativo sempre em 0, deixando a linha de pixels da parte inferior sempre apagada para dar um espaço para a próxima linha). O programa implementado na MCU usa uma biblioteca pronta para o *display*, fazendo sua inicialização e depois escrevendo um caractere “0” na primeira linha e um caractere “1” na segunda linha.

EXERCÍCIO DE INTERFACE SPI:

Os cinco canais que aparecem no arquivo de sinais são os cinco sinais citados na introdução. Note que no início da captura, alguns sinais sofrem rápidas mudanças (“glitches”) e depois se estabilizam. Isto acontece porque neste momento foi dado um “reset” na MCU. A captura é sincronizada com a descida do RST, e por isso este “glitch” aparece.

Depois de aproximadamente 10ms (para estabilização da alimentação do módulo), a inicialização do *display* começa com um “reset” do módulo (RST indo para nível baixo).

1. Qual é o tempo de duração do “reset”? Veja nas páginas 20 e 21 do *datasheet* a duração mínima exigida para o “reset” e diga se o pulso dado pela MCU é suficiente. Observe na seção 8.2 do *datasheet* que após o “reset” vários parâmetros são definidos para manter o *display* desativado, e o conteúdo da DDRAM (que define a ativação de cada bit) é indefinido. Assim, para manter a tela em branco após a ativação do *display*, é importante carregar valores zero em todas as posições da DDRAM.
2. Adicione um analisador SPI. Configure os sinais adequadamente (MOSI, CE e SCK. Note que MISO deve ser ter “none” selecionado. Aqui teremos CPOL = Low e CPHA = Low. Formate os dados apresentados em DI como “Hexadecimal”. Veja a sequência de bytes transmitida logo após a subida do sinal RST. São comandos, pois o sinal DC está em Low. Usando a tabela da página 14 e as informações das páginas 15 e 16 do *datasheet*, explique esta sequência de comandos de inicialização (**faça uma tabela como no exercício prático 2, neste item e no item 4**).
3. Após a inicialização, é transmitida uma grande sequência de bytes em zero com o sinal DC em High. Por que isto é feito?
4. Após a transmissão dos dados em zero, o sinal DC vai novamente a Low e nova sequência de comandos é transmitida. O que estes comandos fazem?
5. Depois desta segunda sequência de comandos, o sinal DC vai novamente a High e dados são transmitidos. Neste momento o caractere correspondente a “0” está sendo desenhado no início da tela. Apresente a sequência de bytes e como eles montam o caractere em uma matriz 5x8 (a última linha sempre com bits em zero).
6. Após o desenho do caractere “0”, há uma nova sequência de comandos e depois de dados para desenhar o caractere “1” no início da segunda linha de texto (a partir da nona linha de pixels). Explique os comandos (**com tabela**) e os dados, como foi feito nos itens 4 e 5.