

Sistemas Elétricos e Eletrónicos de Veículos

Aula Laboratorial nº2

Objetivos:

- Revisões sobre implementação e simulação de circuitos eletrónicos com microcontroladores em software Proteus®;
- Estudo do microcontrolador Microchip® PIC18F46K80;
- Programação de microcontroladores em ambiente Microchip® MPLAB®X IDE / XC8.

1. Implemente o circuito apresentado na Figura 1, com a Microchip® Curiosity HPC Board usando o microcontrolador PIC18F46K80 e o LCD de 2.2" com o controlador ILI9341, em *software* Proteus®. Utilize uma tensão de alimentação de **3.3V para todo o circuito**.
2. Abra o projeto disponibilizado (Moodle da UC): "Aula_2.X" no *software* MPLAB® X IDE da Microchip®, verifique que a compilação do mesmo é realizada sem erros. Resolva os eventuais erros de compilação antes de prosseguir (mantenha as pastas do trabalho o mais próximo da raiz possível "C:\SEEV\Aula_2")
3. Teste o funcionamento do circuito apresentado na Figura 1 com o código disponibilizado. Confirme nas propriedades do projeto, na opção *Proteus VSM Viewer*, tem o projeto do PROTEUS corretamente selecionado. Alternativamente, nas propriedades do microcontrolador no *software* Proteus® selecione o código: "Aula_2.X.production.hex" que aparecerá na diretoria do projeto em caso de compilação sem erros (... \dist\default\production\).

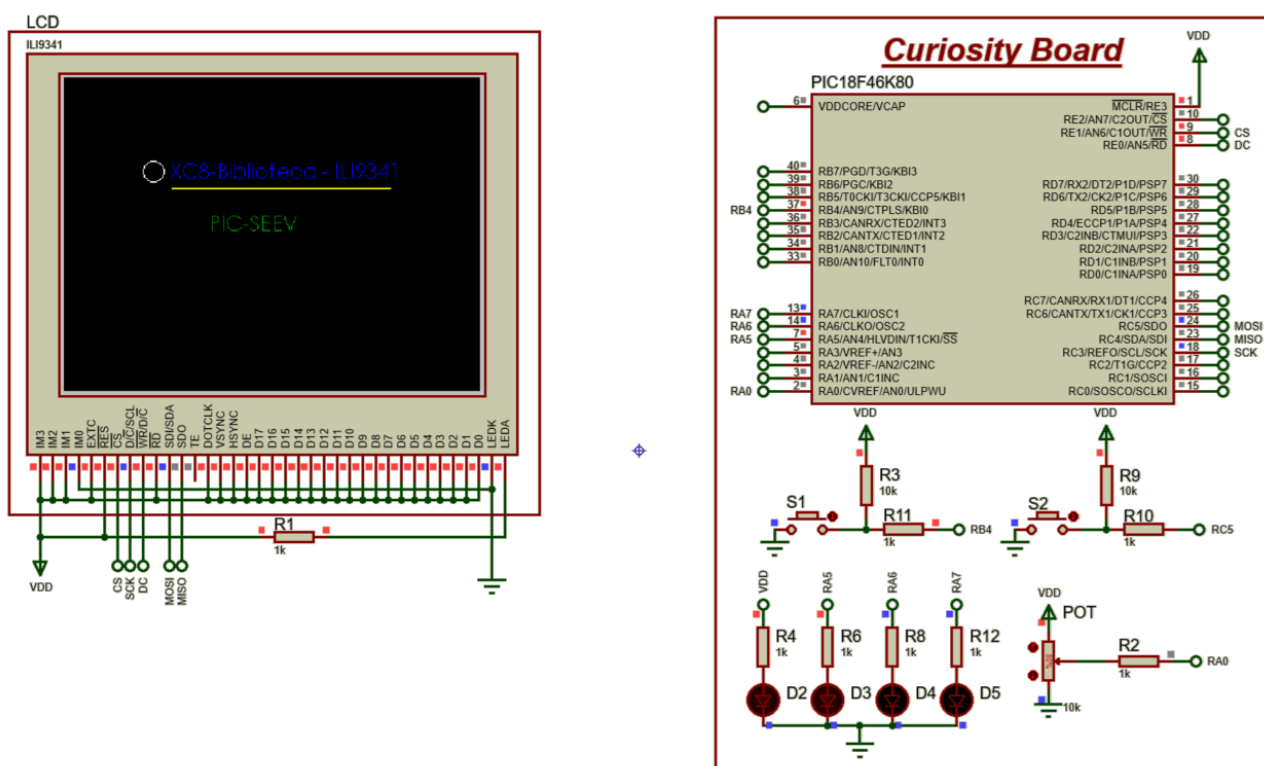


Figura 1

4. Verifique e estude:
 - a. Verifique que o LED D3 acende quando o botão é pressionado.
 - b. Identifique as principais partes do código disponibilizado, e descreva a sua funcionalidade.
 - c. Identifique onde se encontram os valores atribuídos aos registos:
Portos I/O

Oscilador

- d. O que teria de alterar para configurar o relógio para 16MHz.
 - e. Qual a frequência de funcionamento do microcontrolador utilizado?
 - f. Calcule o tempo de execução de uma instrução.
 - g. Desenhe o fluxograma do código disponibilizado.
5. Altere o código apresentado de modo a que o LED apresente um funcionamento com memória (do tipo "toggle button"). Quando o botão é pressionado uma vez o LED é ativo (e permanece ativo), quando o botão é novamente pressionado o LED apaga.

NOTA:

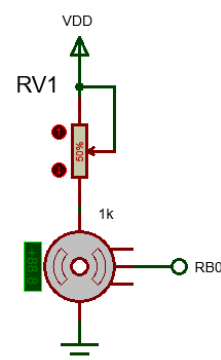
Esta alteração deve ser efetuada recorrendo a interrupções, neste caso será necessário utilizar uma IOC (*Interrupt On Change*) disponível no pino RB4 ao qual o botão S1 se encontra ligado.

--- 2ª Parte ---

6. Adicione ao projeto os ficheiros "lib_ili9341.h" e "lib_ili9341.c" em *Header Files* e *Source Files*, respetivamente. Defina as portas CS e DC do microcontrolador como saídas. Configure a comunicação SPI e "abra" a mesma utilizando *SPI_DEFAULT* como parâmetro da função de "abertura".
7. Explore as funções disponibilizadas pela biblioteca do LCD de modo a conseguir apresentar texto, variáveis e algumas formas geométricas. Tente replicar a Figura 2a imagem no LCD exceto os retângulos 1 e 2 (Resolução de 320x240).
8. Altere o código apresentado de modo a apresentar no LCD, sempre que o botão S2 estiver pressionado, o valor do potenciômetro na escala [0%; 100%]:
Nota: Guarde um *printscreen* como prova de execução.
9. Como mostra a Figura 2b, adicione um motor DC com *encoder* digital conectado ao pino RB0 do microcontrolador. Utilize uma resistência variável para controlar a velocidade do motor e altere as suas propriedades: *Load* = 10, *Effective Mass*=0.0001 e *Pulses per Revolution* = 1.



a.



b.

Figura 2

10. Desenvolva o código necessário para: apresentar no LCD o valor da rotação do motor em RPM's no retângulo 1 da Figura 2a, e o valor do potenciômetro na escala [0%; 100%] no retângulo 2.

- Configure e utilize o temporizador *Timer0*.
- Apresente a fórmula que utilizou para calcular as RPM's em função do tempo por revolução.
- Apresente o fluxograma do programa desenvolvido.
- Utilize interrupções para implementar as funcionalidades pretendidas (poderá fazê-lo depois em estudo autónomo).

Nota: Guarde um *printscreen* como prova de execução.

--- 3ª Parte ---

11. Desenvolva novo código de modo a que o LED D3 acenda sempre que o valor mostrado para a rotação do motor DC seja superior ao valor definido pelo potenciômetro (Exemplo: 80%).
12. Desenvolva novo código de modo a que o LED D3 apresente um funcionamento intermitente (2Hz utilizando um Timer2), sempre que o valor do potenciômetro seja superior a 80. O valor de comparação (80 por omissão) deve ser definido por uma variável.

Referências / documentos recomendados:

[1] Microchip® PIC18F66K80 Family Datasheet (PIC18F46K80)