UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
EA076/EA079
1º. SEMESTRE DE 2014 - TURMAS C, D, E, M, U, W – PROFS. TING, SÉRGIO, CLÉSIO

Roteiro 5 – Periférico *Keypad* e módulo TPMx

1. Objetivos

1. Familiarização com o processamento de interrupções por captura de um evento externo (*Input Capture*) ou por comparação com o contador (*Output Compare*).
2. Familiarização com um circuito de controle de um *keypad*.
3. Familiarização com o módulo TPMx do micro-controlador MCF51CN128.

2. Preparo

Leia referências [2] e [4], consulte ainda [3] e pense numa solução que transforme os estados dos contatos mecânicos das chaves de um *keypad* em um sinal elétrico. Pegue antes do início do experimento um *keypad* no almoxarifado.

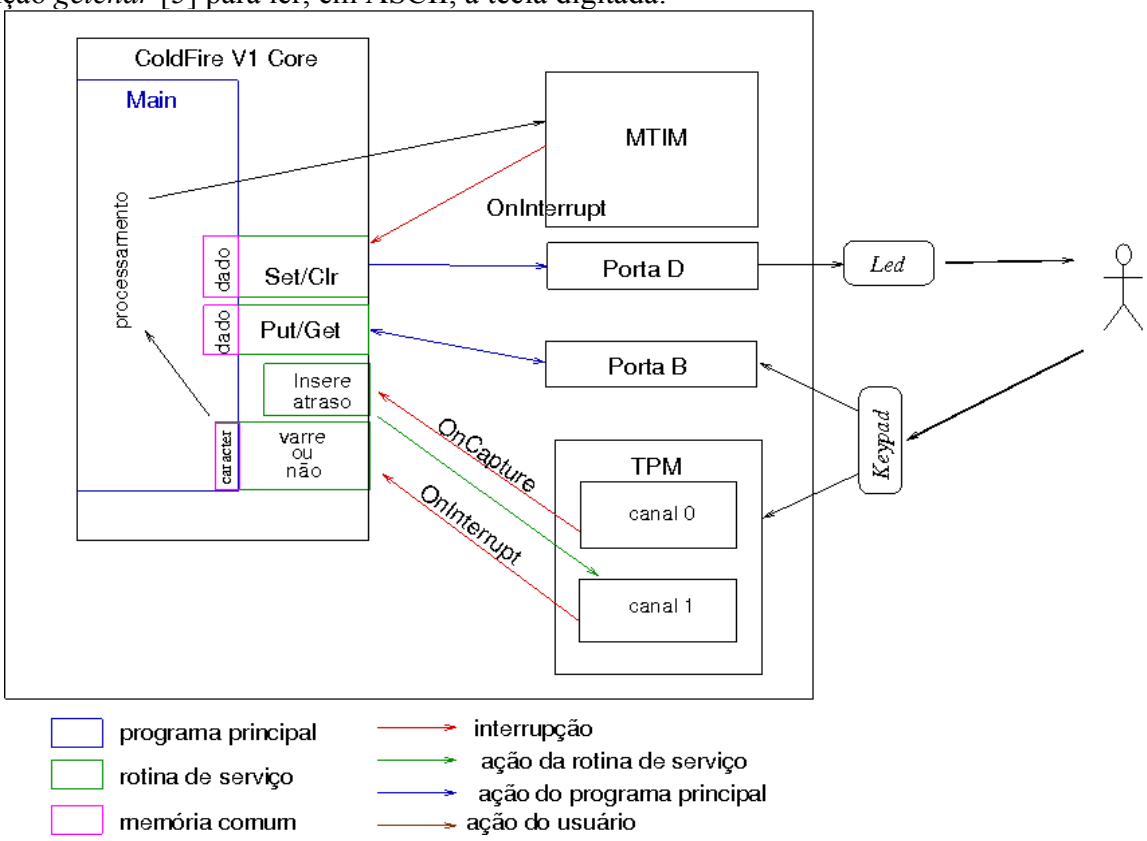
3. Problema

Como seria um circuito de interface entre um *keypad* e o micro-controlador MCF51CN128? Projete uma interface ao circuito de *led* do experimento 2, de forma que o usuário possa controlar a frequência das piscadas do *led* via teclas. As possíveis frequências são 50Hz (1), 40Hz (2), 30Hz (3), 20Hz (4), 10Hz (5), 5Hz (6), 2Hz (7), 1Hz (8), 0.5Hz (9).

Passos Recomendados:

1. Releia [2] com atenção para entender o princípio de conversão dos estados (abre e fecha) de contatos de uma chave em um sinal elétrico. A solução adotada para identificar a chave fechada (função *key_scan*) é por *polling,* varrendo repetitivamente as quatro linhas do *keypad*. Desenhe o esquemático do circuito e adicione-o ao circuito dos experimentos anteriores, ligando as 3 colunas do teclado nos pinos PTB4, PTB5 e PTB6 e as suas 4 linhas nos pinos PTB0, PTB1, PTB2 e PTB3.
2. Com o *Processor Expert* configure a porta B e, com base em [2], implemente a função *key_scan* que capta a entrada de uma tecla. Teste-a em um programa que a chame repetidamente para verificar se todas as teclas são identificadas corretamente.
3. Leia atentamente as Seções 19.1.2, 19.1.3, 19.3 e 19.4.2.1 de [1], a fim de ver uma alternativa para MCF51CN128 capturar um sinal externo (borda de descida ou borda de subida) e gerar com ele uma interrupção. Elabore com base no que você leu e em [3] uma forma de executar a função *key_scan* somente quando uma chave for fechada. Utilizando uma porta lógica AND e o modo *Input Capture* do canal 0 do módulo TPM1, projete o circuito, desenhe o seu esquemático e adicione-o ao circuito anterior.
4. Com o *Processor Expert* configure o módulo TPM1 para que ele reconheça uma mudança no estado dos contatos do *keypad* e gere uma interrupção. Verifique se o seu programa entra na rotina de serviço toda vez que uma tecla for pressionada.
5. Para evitar o fenômeno conhecido por *bouncing* [4], elabore uma alternativa eficiente que introduz um atraso entre o momento da interrupção (passo 3) e o início da varredura (passo 1). No diagrama de blocos apresentado foi dada uma solução utilizando o canal 1 do temporizador TPM1. Integre os passos 1, 3 e 5 num único fluxo de tratamento. Vale observar que *bouncing* ocorre tanto no momento de fechar a chave quanto abrí-la, mas a varredura só deve suceder do *bouncing* correspondente ao

fechamento dos contatos. Teste o seu programa com o seu circuito, implementando por exemplo a função *getchar* [5] para ler, em ASCII, a tecla digitada.



6. Integre o seu circuito e o seu programa ao projeto do roteiro 2.
7. Elabore alguns testes para verificar o comportamento do *led* em relação às teclas digitadas.
8. Documente os códigos com uso do Doxygen.

4. Relatório

A documentação do projeto, no formato dos relatórios anteriores, deve ser postada em PDF no Ensino Aberto (Portfólio dos Grupos, compartilhado apenas com formadores) 24 horas antes do início da próxima aula.

5. Referências Bibliográficas

[1] MCF51CN128 ColdFire® Integrated Microcontroller Reference Manual http://www.freescale.com/files/32bit/doc/ref_manual/MCF51CN128RM.pdf

[2] Keypad Scan http://esd.cs.ucr.edu/labs/decode_key/decode_key.html

[3] Quevedo, A. A. F. e Magalhães, L.P. Descrição do Hardware da Placa Auxiliar ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/complementos/placa-auxiliar-EA870.pdf

[4] The Ganssle Group, A Guide to Debouncing http://www.ganssle.com/debouncing.htm

[5] Getchar http://www.cprogrammingexpert.com/C/Tutorial/io_functions/getchar.aspx