UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E COMPUTAÇÃO

09/05/2016

EA871 - LAB. DE PROGRAMAÇÃO BÁSICA DE SISTEMAS DIGITAIS

EXPERIMENTO 6 – Interrupções

Profª Wu Shin-Ting
Aluno: Tiago Eidi Hatta RA: 160388

Proposta do Experimento

Configuração e programação do MKL25Z128 para processamento de exceções e interrupções, ao controlar as cores e a frequência de piscadas dos leds e a apresentação das informações no LCD da placa auxiliar [8].

Metodologia:

Os seguintes conceitos foram abordados para resolver o problema proposto pelo experimento:

- Interrupção: mecanismo pelo qual componentes distintos do processador (I/O, memória) podem interromper a seqüência normal de execução de instruções do processador. [9]

- Borda de Subida e Descida: a borda de subida é a característica de um sinal que na transição vai de baixo para alto (0 a 1), já borda de descida vai de alto para baixo (1 a 0).

- NVIC: (Nested Vectored Interrupt Controller) controlador de interrupçoes, responsável por monitorar as entradas assíncronas e interromper a execução normal do programa no momento em que a entrada é acionada [3].

- Exceções: mecanismo similar à interrupção, porém a notável diferença é a sua geração por software (provocada pela execução de uma instrução). O processador salva o contexto do processo corrente e transfere o controle para uma rotina predefinida de tratamento da condição ocorrida [9].

- SysTick: Timer do sistema ARM Cortex, que permite um simples contador 24 bits, decrementador, cíclico e resetável com uma operação de escrita (clear-on-write). O seu uso, que é opcional, pode ser utilizado para gerar interrupções periódicas independentes de NVIC [3].

Proposta de Solução

Módulos e Registradores Configurados [1]

Módulo SIM: Especificamente o registrador SCGC5 (System Clock Gating Control Register 5), responsável por habilitar o clock para cada uma das portas relacionadas aos pinos da placa utilizados no experimento.

Módulo PORT: Responsável por dar suporte ao controle das portas da placa principal do kit usado. Registradores:

- PCRn: de acordo com a porta utilizada do pino n, configura a multiplexação para GPIO e o modo de disparo de interrupção de acordo com o tipo de borda do sinal.
Foram configuradas:
- Porta A, pinos 4, 5 e 12: push-buttons da placa auxiliar.
    - Configurados com 0xb em IRQc (11.5.1 em [2])
- Porta B, pinos 18 e 19: leds vermelho e verde.
- Porta D, pino 1: led azul
- Porta C, pinos 0 a 7: Dados GPIO
Os pinos do LCD são ligados ao da porta C da placa.
- Porta C, pinos 8 a 10: RS, E do LCD e LE do latch, respectivamente.

- ISFR: o bit setado corresponde ao pino que detecta (valor lógico 1) ou não (valor lógico 0) uma interrupção configurada.
Foram configuradas:
- Porta A, pinos 4, 5 e 12.

Módulo GPIO: Comunicação digital do processador do MCU. Registradores:
- PDDR: configura os pinos como saída (leds e lcd) ou entrada (push-buttons) para GPIO.
- PDOR: configura os pinos como saída (leds e lcd) ou entrada (push-buttons) para propósitos gerais.

NVIC (Nested Vectored Interrupt Controller): Controlador de Interrupções
NVIC_ISER: habilita as interrupções de acordo com os pinos configurados
NVIC_ICPR: remove estados de interrupção pendentes
- Pino 30: Habilita interrupção Vetor 46, IRQ30
NVIC_IPR7: configura nível de prioridade
- Pinos 16-23: 0xC0 Nível 2 de prioridade

Systick: Timer com contador de 24 bits que permite interrupções periódicas
Registradores configurados:
SYST_CSR: Controla o timer do sistema (habilita ou desabilita o clock do systick) e fornece status do mesmo.
SYST_RVR: Deve ser inicializado com o número do contador a ser reiniciado sempre que o contador chega a zero.
SYST_CVR: Responsável por ler ou limpar o valor do contador.

Abaixo estão os pseudo-códigos das duas partes do experimento, bem como a explicação da solução baseada nos códigos fontes em C.

Pseudo-Código

```
// Funcao responsavel por tratar as interrupcções
IRQ_HANDLER:
        SE pushbutton5 for apertado:
                SE pushbutton5 for solto:
                        Estado_pta5 <- 0
                        SE tempo for maior do que 0
                                nivel = tempo – 1
                        SENAO
                                nivel = 0
                        DESABILITA Clock de Systick
                SENAO
                        Estado_pta5 <- 1
                        tempo = 0
                        HABILITA Clock de Systick
```

```
        SE pushbutton12 for apertado:
                SE pushbutton12 for solto:
                        Estado_pta12 <- 0
                        SE tempo for maior do que 0
                                indice = tempo – 1
                        SENAO
                                indice = 0
                        DESABILITA Clock de Systick
                SENAO
                        Estado_pta12 <- 1
                        tempo = 0
                        HABILITA Clock de Systick
FIM IRQ_HANDLER

// Funcao main
MAIN:
INÍCIO
        CONFIGURE registrador de controle SIM_SCGC5[10] de forma a habilitar clock do módulo GPIO
        CONFIGURE registrador de controle PORTB_PCR18[8:10] de forma que o pino 18 da porta B
        sirva GPIO
        CONFIGURE registrador de controle PORTE_PCR23[8:10] de forma que o pino 23 da porta E
        sirva GPIO

        CONFIGURE registrador de controle GPIOB_PDDR[18, 19] de forma que os pinos 18 e 19 da
        porta B sejam de saída
        CONFIGURE registrador de controle GPIOE_PDDR[23] de forma que o pino 23 da porta E seja
        de saída

        CONFIGURE registrador de controle GPIOC_PDDR[0:10] de forma que os pinos 0 a 10 da porta
        C sejam de saída

        CONFIGURE registrador de controle GPIOA_PDDR[4, 5, 12] de forma que os pinos 4, 5 e 12 da
        porta A sejam de entrada
        CONFIGURE registrador de controle GPIOA_ISFR[4, 5, 12] de forma que os pinos 4, 5 e 12 da
        porta A tenham interrupções detectadas

        CONFIGURE NVIC_ISER[30]
        CONFIGURE NVIC_ICPR[30] para habilitar a interrupção pelo Vetor 46, IRQ30
        CONFIGURE NVIC_IPR7[16:23] configura nível de prioridade com o valor de 0xC0

        CONFIGURE registrador Systick SYST_CSR
        CONFIGURE registrador Systick SYST_RVR para ciclo de 0.25 s
        CONFIGURE registrador Systick SYST_CVR desabilitando o clock

        INICIA LCD DE ACORDO COM O INDICADO PELO FABRICANTE

        CRIA CARACTERE ê (chama criaCaractere)

        PARA LOOP VERDADEIRO
                SE estado_pta5 e estado_pta12 são nulos
                        MANDA STRING PARA O VISOR: “Cor: “
                        MANDA STRING PARA O VISOR: NOME DA COR SELECIONADA DO LED
                        MANDA STRING PARA O VISOR: “Frequ”
                        MANDA STRING PARA O VISOR: “ê”
                        // (criado anteriomente e armazenado em CGRAM)
                        MANDA STRING PARA O VISOR: “ncia: ”
                        MANDA STRING PARA O VISOR: FREQUENCIA CONVERTIDA EM ASCII
```

// as interrupçoes podem mudar o valor de “nivel” ou de acordo com IRQ_HANDLER

ACENDE LED DE ACORDO COM O VALOR DE INDICE

TEMPO: INVERSO DA FREQUENCIA SELECIONADA POR NIVEL

DELAY10US passado como parametro TEMPO determinado de delay

APAGA LED

CONVERTE PARA ASCII O VALOR FLOAT DA FREQUENCIA DE PISCADA

DELAY10US passado como parametro TEMPO determinado de delay

FIM PARA

FIM

Código Fonte

Enviado como anexo ao relatório pelo Ensino Aberto. Documentado de acordo com o especificado pela Doxygen.

Testes

Para os testes, foram realizadas as seguintes etapas:

1) Mudança do conteúdo no visor do led de acordo com o pushbutton 12 da placa auxiliar [8]
2) Mudança do conteúdo do visor e cor dos leds de acordo com o pushbutton 12
3) Mudança do conteúdo do visor, cor e frequência de piscadas dos leds de acordo com pushbutton 5 e 12, respectivamente

Conclusão

A interrupção é um mecanismo essencial na programação de qualquer microcontrolador. De forma eficiente, consegue interagir com o usuário em um tempo de resposta mais rápido, a partir de entradas assíncronas do que se fosse utilizado o polling, por exemplo. O timer systick é um dos principais recursos que permitem isso, pois seu funcionamento é mais preciso do que utilizando uma função criada pelo programador, já que está atrelada diretamente ao hardware. Para o experimento, decidiu-se por opção enquanto o usuário estiver apertando o botão, suspender a piscada do led, para que que, ao se soltar o pushbutton, a mudança desejada fosse instantânea. Percebeu-se que com a frequência de 0.5 Hz, havia um delay ao se alterar a cor do led para preto. Isso acontecia pois, no loop da função main, apesar da variável “indice” ser alterada, se o programa estivesse executando entre o delay das piscadas, haveria um tempo (consideravelmente grande para a percepção humana, de 2 segundos) a ser passado até que o estado dos leds fosse alterado.

A conclusão final é que o uso de interrupções é importante para um projeto robusto e que atenda às demandas do usuário principalmente relacionadas ao tempo de

resposta. Para isso, a existência de diferentes formas de configuração, como em relação aos níveis de prioridade, a conexão de entrada/saída com os pinos do MCU, e o uso mais preciso do clock de systick, por exemplo, permitem uma flexibilidade para resolver uma gama considerável de problemas com o MKL25Z128.

Referências

[1] FRDM-KL25Z User's Manual.

ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/ARM/FRDMKL25Z.pdf

[2] KL25 Sub-Family Reference Manual

ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/ARM/KL25P80M48SF0RM.pdf

[3] ARMv6-M Architecture Reference Manual – ARM Limited.

ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/ARM/ARMv6-M.pdf

[4] EA871 – Descrição do Hardware da Placa Auxiliar.

ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/ARM/DescricaoDoHardware.pdf

[5] LCD Tutorial for Interfacing with Microcontrollers. Rickey's World.

http://www.8051projects.net/lcd-interfacing/index.php

[6] Datasheet do display LCD.

ftp://ftp.dca.fee.unicamp.br/pub/docs/ea079/datasheet/AC162A.pdf

[7] TABELA ASCII – UFRJ

http://equipe.nce.ufrj.br/adriano/c/apostila/tabascii.htm

[8] EA871 - Descrição do Hardware da Placa Auxiliar

ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/ARM/DescricaoDoHardware.pdf

[9] Arquitetura e Organização de Computadores - 8ª Ed. - William Stallings (Cap. 3)