

EA871 – LAB. DE PROGRAMAÇÃO BÁSICA DE SISTEMAS DIGITAIS

EXPERIMENTO 4 – Linguagem de Montagem (Assembly) - II

Profa. Wu Shin-Ting

OBJETIVO: Desenvolver um projeto em linguagem C e linguagem de montagem

ASSUNTO: Linguagem de montagem dentro dos programas em linguagem C

O que você deve ser capaz ao final deste experimento?

Saber integrar os códigos em linguagem de montagem nos códigos em linguagem C.

INTRODUÇÃO

Vimos no roteiro 3 que programando em linguagem de montagem temos total controle sobre o tempo e a quantidade de memória que um programa necessita na sua execução. Neste experimento vamos aplicar o que você aprendeu para projetar uma subrotina que gere um atraso especificado com precisão de microsegundos dentro de um projeto em linguagem C.

EXPERIMENTO

1. *Vamos entender como se integra códigos em linguagem de montagem num programa escrito em C? Leia* atentamente o [documento](#) [3] para ver como se faz esta integração.
2. *Qual é o tempo gasto na execução de um programa?* Veja na Seção 3.2, p. 26, em [1] que os ciclos de instrução de cada instrução do repertório de Cortex-M0+ são especificados em termos de ciclos de relógio do processador. Sabendo que a frequência de operação do núcleo do nosso microcontrolador está configurado para 20.971520 MHz, podemos estimar o tempo de execução de um programa somando os ciclos de cada instrução e multiplicar a soma pelo período do relógio do processador. Note, porém, que esta é uma estimativa grosseira pois há otimizações que reduzem este tempo estimado.
3. *Vamos colocar em prática?* Substitua o código da rotina *delay* no programa [apostila.c](#) pela rotina *void delay10us(int t)*, de forma que a execução da rotina *delay* seja em $t * 0.00001s$.
 - a) Como se passa o valor do argumento *t* de uma chamada em C para as instruções em *assembly* dentro da função *delay10us(int t)*?
 - b) Elabore um programa em *assembly* equivalente ao laço de espera da função *delay* em C.
 - c) Veja na Seção 3.2, p. 26, em [1] os ciclos de instrução de cada instrução do repertório de Cortex-M0+ e estime o tempo necessário para executar uma iteração. Quantas iterações seriam necessárias para um período de 10us? E para um período *t*? Vale a pena lembrar que a instrução de desvio também gasta tempo (p. 28 em [1]).
 - d) O acesso ao sinal do pino PTB18, que alimenta o *led* vermelho, não é fácil. No Anexo A a ponta de prova do osciloscópio foi colocado no pino PTE23 da placa auxiliar [2]. Para que o sinal PTB18 seja “replicado” no pino PTB23, basta programar (em C) o pino PTE23 de forma similar ao pino PTB18. Vamos adicionar a configuração do pino PTE23 no programa [apostila.c](#).
 - e) Verifique a acurácia do seu código com os seguintes valores de *t* com uso de osciloscópio conforme a imagem no Anexo A: 1, 10, 100, 1000, 10000 e 50000.
4. Substitua a rotina *delay* do seu programa de pisca-pisca de cor branca do roteiro 2 pela nova função *delay10us* e modifique o seu projeto para que os *leds* pisquem na frequência de 1Hz.

RELATÓRIO

Elabore o relatório sobre o projeto de pisca-pisca com cores variadas na frequência de 1Hz. O relatório deve conter a identificação completa, especificação objetiva do problema, o pseudo-código da solução do problema, incluindo a inicialização dos registradores de controle do microcontrolador, os testes de validação do seu código de atraso, incluindo as imagens da tela do osciloscópio para diferentes frequências, e o código-fonte documentado de acordo com a sintaxe de Doxygen e exportado no formato de compressão *.zip. Dica: A instrução NOP não faz nada, só gasta tempo correspondente a um ciclo de relógio do núcleo. (veja Seção A6.7.47 em [1])

REFERÊNCIAS

Todas as referências podem ser encontradas nos links abaixo ou ainda na página do curso.

[1] Cortex-M0+ Technical Reference Manual (Revision: r0p0)

<ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/ARM/Cortex-M0+.pdf>

[2] Wu Shin-Ting e A.A.F. Quevedo. Ambiente de Desenvolvimento – Hardware

ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/apostila_C/AmbienteDesenvolvimentoHardware.pdf

[3] Wu Shin-Ting. Linguagem de Montagem

ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/apostila_C/LinguagemMontagem.pdf

Baseado nos Roteiros dos Experimentos 3 e 4 do Segundo Semestre de 2015 (Agosto de 2016)
Revisado em Fevereiro de 2017

ANEXO A: PONTOS DE ACESSO A PTE23 e TERRA



Fonte: Roteiro do Experimento 4 do Segundo Semestre de 2015 (<http://faraj7.github.io/>).