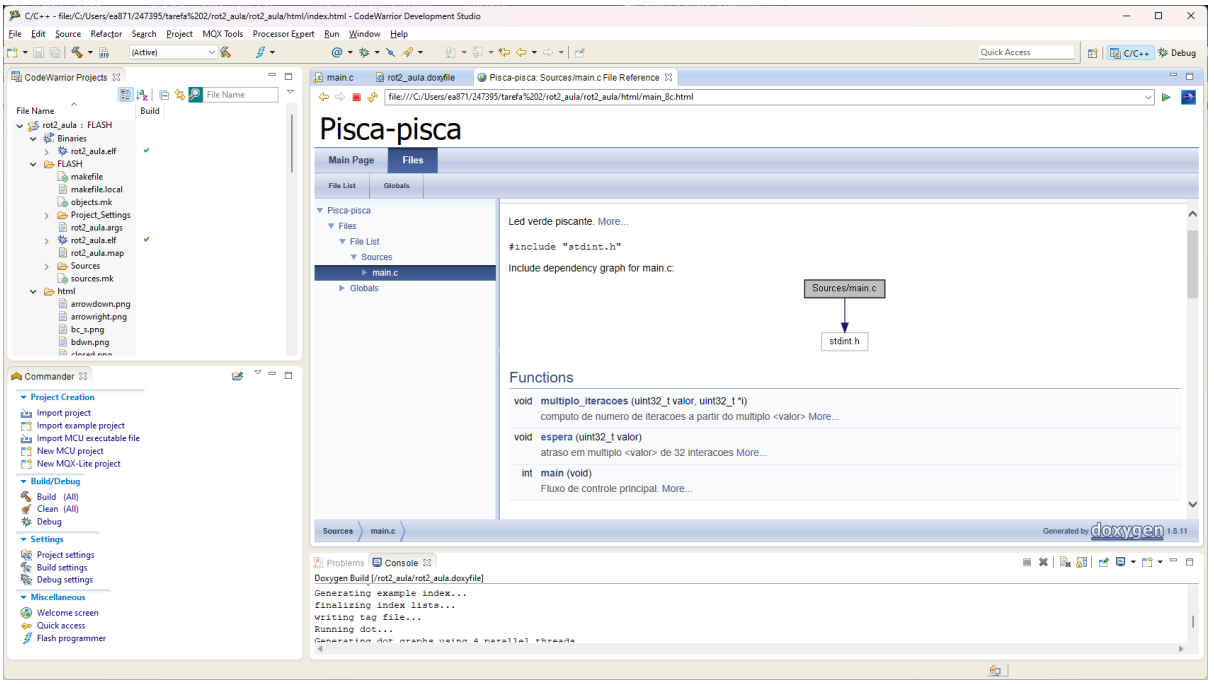
FEEC - UNICAMP;
EA871;
EXPERIMENTO DA TAREFA 2;
VINÍCIUS ESPERANÇA MANTOVANI, RA: 247395;

ÍTEM 1:

a) Após a transferência do .elf, o módulo inserido em MODULES na perspectiva debug é o “rot2_aula.elf”.
b) Ao executar o .elf, percebe-se que um led pertencente a placa pisca periodicamente
c)



d) O espaço de memória requerido para as instruções do projeto é de 1088 bytes. Já data requer um espaço de 28 bytes e bss requer um espaço de 2084 bytes.

ÍTEM 2:

a) Inicialmente, temos uma função em __thumb_startup que corresponde com a sequência de inicialização exposta no livro, a função __init_registers, responsável por inicializar os registradores. Então, é executada a função __copy_rom_sections_to_ram que é coincidente com o passo da seção 1.1.4.2.2, ou seja, corresponde à inicialização da RAM, com cópia das funções da FLASH para a RAM. Por fim, outra função coincidente com a sequência da seção 1.1.4 é exit(main(__argc_argv(__MAX_CMDLINE_ARGS, argv), argv)), que chama a main do programa.
b) main: m = 0x88; multiplo_interacoes: n = 0x1a; espera: p = 0x34.
c) As instruções são todas de 2 bytes, com exceção das instrução “bl”, que tem tamanho de 4 bytes.
d) m_text é armazenada na memória FLASH, enquanto que m_data é armazenada na RAM do aparelho. Já a pilha, está armazenada na memória RAM, também.

e)

	Endereços pré-fixados (registradores)	locais	globais
main	- SIM_SCGC5 0x40048038u - PORTB_PCR19 0x4004a04c - GPIOB_PDDR 0x400ff054u - GPIOB_PCOR 0x400ff048u - GPIOB_PSOR 0x400ff044u		- uint8_t teste
multiplo_iteracoes		- uint32_t valor - uint32_t *i	
espera		- uint32_t valor - static uint32_t contador_offset - static uint16_t contador - uint32_t i	

f) Caso com todas as variáveis incluídas: data = 28 bytes; bss = 2084 bytes.
Caso com variável contador excluída: data = 28 bytes; bss = 2080 bytes.
Caso com variável contador e contador offset excluídas: data = 24 bytes; bss = 2080 bytes.
Caso com as três variáveis excluídas: data = 24 bytes; bss = 2076 bytes.

As diferenças nos números de bytes em data e bss devem ser explicadas separadamente. Isso porque, a diferença causada pela remoção de contador_offset ocorre no número de bytes em data, uma vez que essa variável tem seu valor definido logo que é declarada. Já as variáveis teste e contador, alteram o número de bytes em bss, pois teste não tem valor determinado ao ser declarada e, contador é declarada e definida como 0, logo se enquadra, também, na sessão bss.

ÍTEM 3:

a) 0x00000000 contém inicialmente o valor 0x20003000 e o endereço 0x00000004 contém inicialmente o valor 0x0000085D. Enquanto que SP_MAIN contém 0x20003000 e PC contém 0x0000085c. A diferença entre o valor em PC e o valor armazenado no endereço 0x00000004 ocorre por conta do bit menos significativo no endereço 0x00000004 ser um identificador de thumb, tendo o valor 1, enquanto que em pc o valor já é o certo, 0x0000085c.

b) Os bytes são armazenados do menos significativo ao mais significativo, ou seja, é dado em little-endian.

00000000	00	30	00	20	5D	08	00	00

c) Antes de iniciar: SIM_SCGC5 → 0x00000180; PORTB_PCR19 → 0x00000000; GPIOB_PDDR → 0x00000000; GPIOB_PCOR → non-readable; GPIOB_PSOR → non-readable.
alterações:
1°: SIM_SCGC5 → 0x00000580; PORTB_PCR19 → 0x00000005;
2°: PORTB_PCR19 → 0x00000105;
3°: GPIOB_PDDR → 0x00080000;
Nota-se que, embora GPIOB_PCOR e GPIOB_PSOR tenham assumido valor non-readable desde o início, GPIOB_PDOR e GPIOB_PDIR variam de valor entre 0x00000000 e 0x00080000 sempre que são alterados GPIOB_PCOR e GPIOB_PSOR no loop da função main.

d) Com a pilha no lugar padrão:
- main: teste → 0x1ffff020

- espera:
+ valor → 0x20002fe4
+ contador_offset → 0x1ffff000
+ i → 0x20002fec
+ teste → 0x1ffff020

- multiplo_iterações:
+ valor → 0x20002fd4
+ i → 0x20002fd0
+ teste → 0x1ffff020

Com a pilha alterada para 0x20002800:
- main: teste → 0x01ffff20

- espera:
+ valor → 0x200027e4
+ contador_offset → 0x1ffff000
+ i → 0x200027ec
+ teste → 0x1ffff020

- multiplo_iterações:
+ valor → 0x200027d4
+ i → 0x200027d0
+ teste → 0x1ffff020

Analisando esses resultados, notamos que as variáveis locais são armazenadas em endereços próximos ao topo da pilha, enquanto que as variáveis globais são armazenadas em endereços mais próximos da base da pilha, bastante distantes das locais.

e) O fluxo é desviado para o endereço 0x0000085c, enquanto que na aba 0x0000085d. Inicialmente, são inicializados os registradores pela função __init_registers, em seguida é feita a inicialização de hardware por __init_hardware. Então, a sessão .bss é preenchida com zeros e as funções são copiadas da ROM para RAM (__copy_rom_sections_to_ram). Por fim, segue-se com inicializações padrão de C e de usuário. Essa inicialização está de acordo com a recomendação do fabricante, embora haja passos suprimidos no desenrolar do codewarriors como vimos na questão 2a.
   Os endereços efetivos das instruções mostrados na Figura 1 do roteiro são alterados de acordo com o novo começo da sessão m_text.
f) Ao alterar o bit 19 do registrador GPIOB_PDOR para o valor 0, notamos que o led acende, enquanto que, ao colocar o valor 1, o led apaga.