

Disciplina EL9630 / NEA630

Experimento 6

Desenvolvimento e Simulação de Programas em Linguagem C Utilizando Interrupções.

Objetivo:

Estamos estudando os dispositivos RL78 e a ferramenta de software IAR, contidos nos capítulos do livro texto do Fábio Perreira, disponível em:

<https://www.renesas.com/br/en/img/misc/documents/books/microcontroladores-rl78-guia-basico.pdf>

Na aula de hoje de laboratório será sobre o Desenvolvimento e Simulação de Programas Utilizando Interrupções.

O objetivo desta Experiência é Desenvolver, Simular e Analisar Programas utilizando as Interrupções dos dispositivos RL78.

Parte prática:

Inicialize o software IAR Embedded Workbench. Realize passo a passo o processo de simulação do Programa.

Use o programa 6.3.1. Exemplo contido na página 132 do livro Microcontroladores-RL78-Guia-Basicocomo.

Será usado o vetor de interrupção 3 e o vetor de interrupção 4, disponíveis nos RL78/G13 para as interrupções externas INTP1 e INTP2.

| Número do Vetor | Endereço | Nome | Flag | Máscara | Fonte |
|-----------------|----------|--|---|---|--|
| - | 0x00000 | RST POR LVD WDT TRAP IAW RPE | - - RESF:LVIRF RESF:WDTRF RESF:TRAP RESF:IAWRF RESF:RPERF | - - - - - IAWCTL:IAWEN RPECTL:RPERDIS | Pino de reset POR LVD Watchdog Opcode ilegal Acesso ilegal a memória Erro de paridade da RAM |
| - | 0x00002 | DBG | - | - | On chip debugger (OCD) |
| 0 | 0x00004 | INTWDTI | IFOL:WDTIIF | MKOL:WDTIMK | Watchdog |
| 1 | 0x00006 | INTLVI | IFOL:LVIIF | MKOL:LVIIMK | Deteção de baixa tensão |
| 2 | 0x00008 | INTP0 | IFOL:PIF0 | MKOL:PMK0 | Pino INTP0 |
| 3 | 0x0000A | INTP1 | IFOL:PIF1 | MKOL:PMK1 | Pino INTP1 |
| 4 | 0x0000C | INTP2 | IFOL:PIF2 | MKOL:PMK2 | Pino INTP2 |

Quando ocorrer a interrupção INTP1 acenderá o Led, e quando ocorrer a interrupção INTP2 apagará o Led.

Programa a ser simulado. “Experiência 6”

```
#include "ior5f1001e.h"
#include "ior5f1001e_ext.h"
#include "intrinsics.h"
#include "myRL78.h"

// Configura watchdog = desligado
#pragma location = "OPTBYTE"
__root __far const char opbyte0 = WDT_OFF;
// Configura detector de baixa tensão = desligado
#pragma location = "OPTBYTE"
__root __far const char opbyte1 = LVD_OFF;
// oscilador 32MHz flash high speed
#pragma location = "OPTBYTE"
__root __far const char opbyte2 = FLASH_HS | CLK_32MHZ;
// debug ativado, com apagamento em caso de falha de autenticação
#pragma location = "OPTBYTE"
__root __far const char opbyte3 = DEBUG_ON_ERASE;
/* Configura security ID */
#pragma location = "SECUID"
__root __far const char senha[10] = {0,0,0,0,0,0,0,0,0,0};

#define LED P6_bit.no3

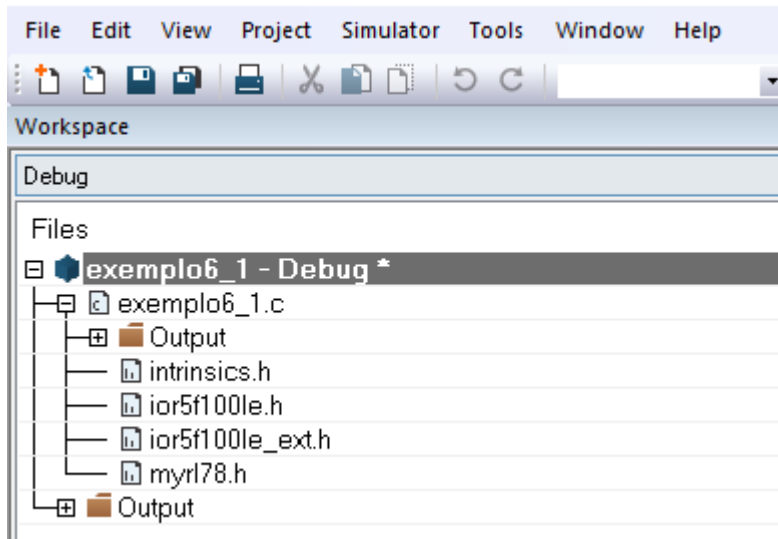
#pragma vector = INTP1_vect
__interrupt void trata_INTP1(void)
{
    LED = 0; // acende led
}

#pragma vector = INTP2_vect
__interrupt void trata_INTP2(void)
{
    LED = 1; // apaga led
}

void main(void)
{
    PM6_bit.no3 = 0; // configura LED como saída
    EGN0 = BIT2 | BIT1; // INTP1 e INTP2 na borda de descida
    PIF1 = 0; // apaga flag da INTP1
    PIF2 = 0; // apaga flag da INTP2
    PMK1 = 0; // habilita INTP1
    PMK2 = 0; // habilita INTP2
    LED = 1; // led desligado
    __enable_interrupt(); // habilita interrupções globais
    while (1);
}
```

Criar Projeto:

- 1) Crie um novo projeto, e digite o código do programa mostrado acima



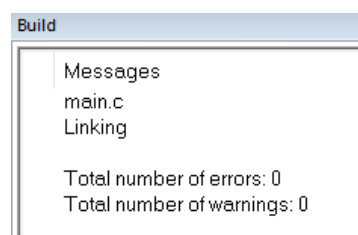
Compile o Projeto:

- 2) Configure o compilador para **não** realizar otimização de código para que seja simulado exatamente o que foi programado.

Compile o projeto.

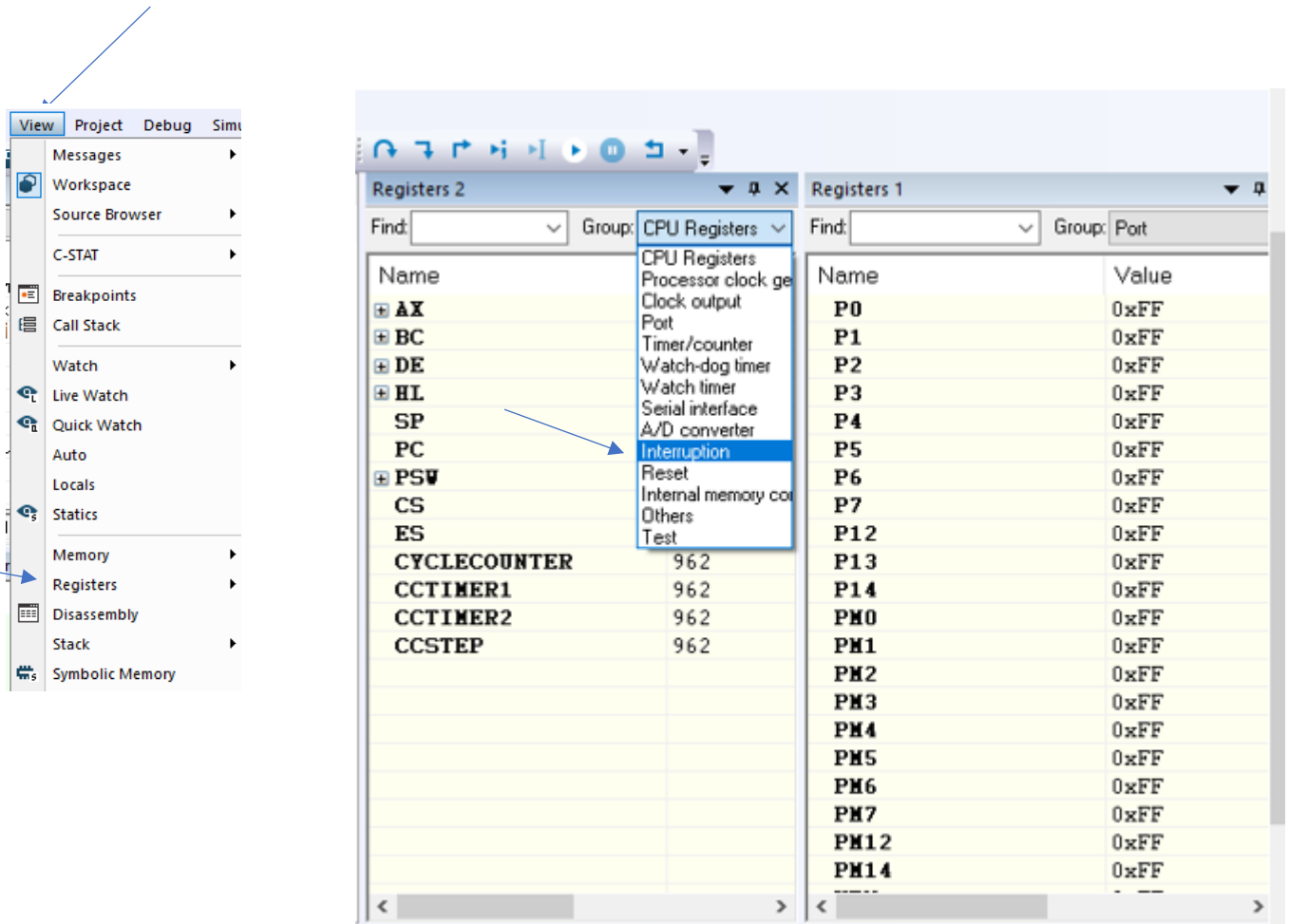


Observe se não ocorreu nenhum erro



Simulação:

- 3) Entre em View Register e adicione na tela do simulador dois Registers, um para visualização do Port P6, e o outro para visualização da Interrupção.



Verifique o estado do registrador IF0L o qual possui informações da ocorrência das interrupções.

| Registers 2 | |
|-------------|---------------------|
| Find: | Group: Interruption |
| Name | Value |
| EGP0 | 0xFF |
| EGN0 | 0xFF |
| EGP1 | 0xFF |
| EGN1 | 0xFF |
| + IF2 | 0xFFFF |
| + MK2 | 0xFFFF |
| + PR02 | 0xFFFF |
| + PR12 | 0xFFFF |
| - IF0 | 0xFFFF |
| + IF0H | 0xFF |
| - IF0L | 0xFF |
| PIF5 | 1 |
| PIF4 | 1 |
| PIF3 | 1 |
| PIF2 | 1 |
| PIF1 | 1 |
| PIF0 | 1 |
| LVIIIF | 1 |
| WDTIIF | 1 |
| - IF1 | 0xFFFF |
| + IF1H | 0xFF |

Adicione dois Breakpoints, um no acende Led e outro no apaga Led .

```

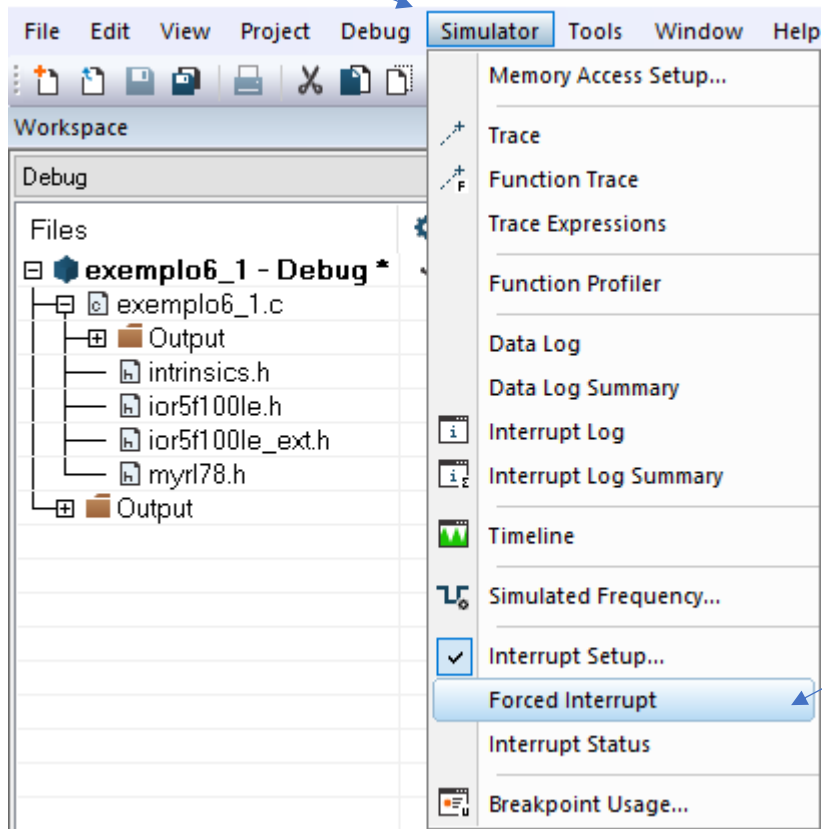
exemplo6_1.c x
main()

#pragma vector = INTP1_vect
__interrupt void trata_INTP1(void)
{
    LED = 0; // acende led
}

#pragma vector = INTP2_vect
__interrupt void trata_INTP2(void)
{
    LED = 1; // apaga led
}

```

Em Simulator selecione Forced Interrupt



| Forced Interrupt | |
|------------------|---|
| Interrupt | Description |
| RST | 0x00 - |
| INTDBG | 0x02 - |
| INTWDTI | 0x04 0 MK0.MK0L.WDTIMK IF0.IF0L.W... |
| INTLVI | 0x06 1 MK0.MK0L.LVIMK IF0.IF0L.LVIIF... |
| INTP0 | 0x08 2 MK0.MK0L.PMK0 IF0.IF0L.PIF0 ... |
| INTP1 | 0x0A 3 MK0.MK0L.PMK1 IF0.IF0L.PIF1 ... |
| INTP2 | 0x0C 4 MK0.MK0L.PMK2 IF0.IF0L.PIF2 ... |
| INTP3 | 0x0E 5 MK0.MK0L.PMK3 IF0.IF0L.PIF3 ... |

USE O BOTÃO DA DIREITA DO MOUSE PARA FORÇAR A INTERRUPÇÃO

Execute passo a passo o programa e verifique a ocorrência da Interrupção pelos Breakpoints e em Registers 2.

| Registers 2 | |
|-------------|---------------------|
| Find: | Group: Interruption |
| Name | Value |
| EGP0 | 0xFF |
| EGN0 | 0x06 |
| EGP1 | 0xFF |
| EGN1 | 0xFF |
| + IF2 | 0xFFFF |
| + MK2 | 0xFFFF |
| + PR02 | 0xFFFF |
| + PR12 | 0xFFFF |
| - IF0 | 0xFFEF |
| + IF0H | 0xFF |
| - IF0L | 0xEF |
| PIF5 | 1 |
| PIF4 | 1 |
| PIF3 | 1 |
| PIF2 | 0 |
| PIF1 | 1 |
| PIF0 | 1 |
| LVIIF | 1 |
| WDTIIF | 1 |
| - IF1 | 0xFFFF |
| + IF1H | 0xFF |

```

trata_INTP1()
{
    #define LED P6_bit.no3

    #pragma vector = INTP1_vect
    __interrupt void trata_INTP1(void)
    {
        LED = 0; // acende led
    }

    #pragma vector = INTP2_vect
    __interrupt void trata_INTP2(void)
    {
        LED = 1; // apaga led
    }
}

```

Altere a ocorrência da interrupção Forced Interrupt.

Relatório:

Para a entrega do relatório deverá conter:

- A) Realize a simulação do Programa Exemplo, mostrado na página 3 deste relatório, e explique o seu funcionamento. Elabore também um fluxograma.

- B) Opcional: Alguns prints das telas do simulador das tarefas realizadas.