

Instituto Tecnológico de Aeronáutica - ITA

Laboratório

3

LUCAS NICOLLI TOSI

Sembro de 2019





1 Objetivo

1.1 Objetivo deste laboratório

Implementação de mecanismos de sincronização entre tarefas utilizando máquinas de estados para as tarefas do RTOS acessando recursos compartilhados.

2 Desenvolvimento

2.1 Desenvolvimento do laboratório

Os seguintes itens deverão ser exercitados neste laboratório:

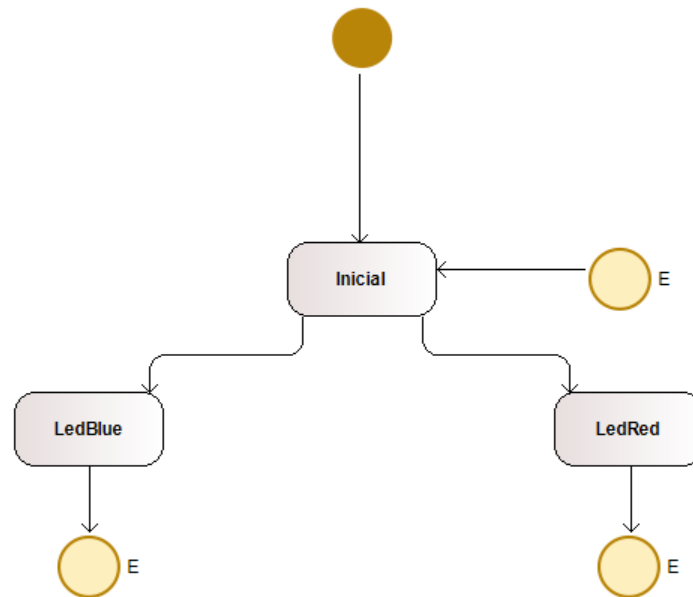
1. Utilização dos modelos (SysML), máquinas de estados e tarefas do laboratório anterior.
 - Utilizar mecanismos de sincronização para acesso ao controle do comando de cores em LED RGB.
2. Utilizar semáforos num primeiro momento, parte A.
 - OSA_SemaCreate
 - OSA_SemaWait
 - OSA_SemaPost
3. Utilizar *mutex* num segundo momento, parte B.
 - OSA_MutexCreate
 - OSA_MutexLock
 - OSA_MutexUnlock
4. Realizar a comparação dessas duas implementações diferentes, ou seja, partes A e B.
5. Utilizar *delays* para controle de tempo das tarefas, se necessário.

3 Referências

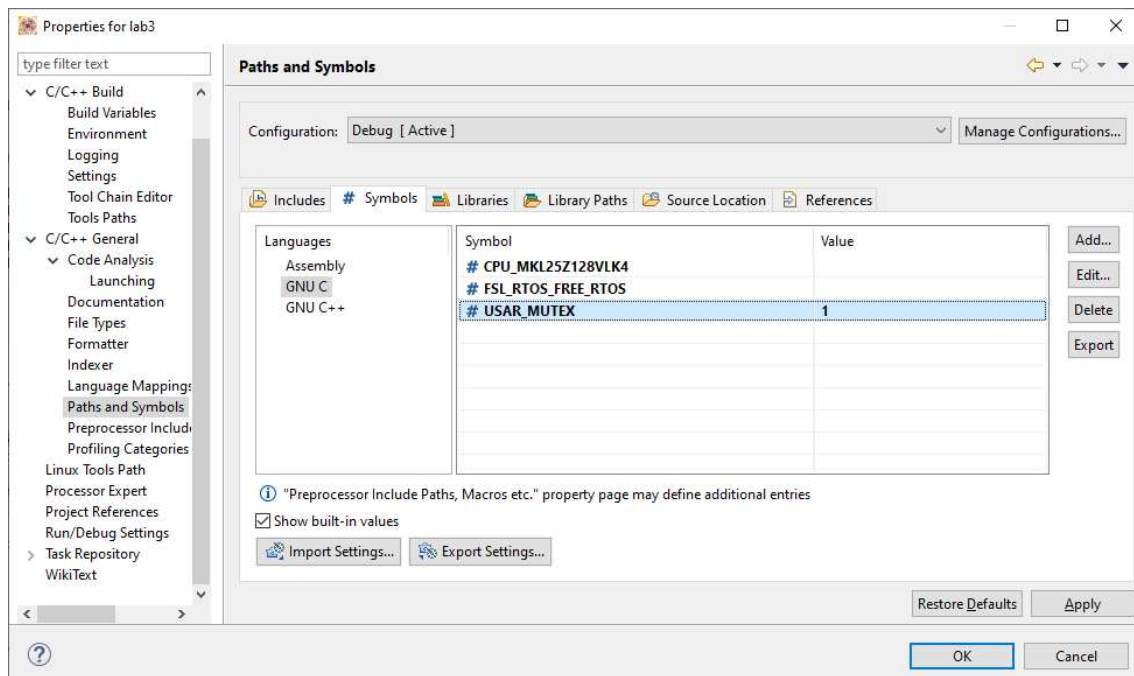
Notas de aula baseadas nos seguintes textos:

- [1] Freescale Semiconductor, *KSDK12APIRM - Kinetis SDK v.1.2 API Reference Manual*, 0 ed., 2015.
- [2] Real Time Engineers Ltd., “Freertos - <http://www.freertos.org>.”

1. State Machine Diagram:



2. Criada, nas Propriedades do Projeto, a macro **USAR_MUTEX** para realização do Lab3. Com isso, a compilação pôde ser controlada para utilizar, ora o Semáforo, ora o Mutex:



3. Código da Task 1, no arquivo os_task.c:

```
void Task1_task(os_task_param_t task_init_data)
{
    /* Write your local variable definition here */

    uint8_t estadoLedAzul=0;

#ifdef PEX_USE_RTOS
    while (1) {
#endif

        /* Write your code here ... */

        /*OSA_TimeDelay(10);*/

        /* Example code (for task release) */

        switch (estadoLedAzul){
            case 0:
                if USAR_MUTEX==1
                    if(OA_MutexLock(&mutexrgb,2000)){
                #else
                    if (OSA_SemaWait(&semargb,2000) != kStatus_OSA_Success){
                #endif
                    estadoLedAzul=3;
                }
                ledrgb_setBlueLed();
                estadoLedAzul=1;
                OSA_TimeDelay(1000);
                break;
            case 1:
                ledrgb_clearBlueLed();
                if USAR_MUTEX==1
                    if(OA_MutexUnlock(&mutexrgb)){
                #else
                    if (OSA_SemaPost(&semargb) != kStatus_OSA_Success){
                #endif
                    estadoLedAzul=3;
                }
                estadoLedAzul=0;
                OSA_TimeDelay(1000);
                break;
            case 2:
                break;
            case 3:
                break;
        }

#ifdef PEX_USE_RTOS
    }
#endif
}
```

4. Código da Task 2, no arquivo os_task.c:

```
void Task2_task(os_task_param_t task_init_data)
{
    /* Write your local variable definition here */

    uint8_t estadoLedVermelho=0;

#ifdef PEX_USE_RTOS
    while (1) {
#endif

        /* Write your code here ... */

        /* OSA_TimeDelay(10); */

        /* Example code (for task release) */

        switch (estadoLedVermelho){
            case 0:
                ledrgb_clearRedLed();
#if USAR_MUTEX==1
                if(OSA_MutexUnlock(&mutexrgb)){
#else
                if (OSA_SemaPost(&semargb) != kStatus_OSA_Success){
#endif
                    estadoLedVermelho=2;
                }
                estadoLedVermelho=1;
                OSA_TimeDelay(1000);
                break;
            case 1:
#if USAR_MUTEX==1
                if(OSA_MutexLock(&mutexrgb,2000)){
#else
                if (OSA_SemaWait(&semargb,2000) != kStatus_OSA_Success){
#endif
                    estadoLedVermelho=3;
                }
                ledrgb_setRedLed();
                estadoLedVermelho=0;
                OSA_TimeDelay(1000);
                break;
            case 2:
                break;
            case 3:
                break;
        }

#ifdef PEX_USE_RTOS
    }
#endif
}
```

5. Arquivo `rtos_main_task.c`, com a rotina de inicialização e declaração de variáveis:

```
#if (USAR_MUTEX==1)
    mutex_t mutexrgb=NULL;
#else
    semaphore_t semargb=NULL;
#endif
```

```
void main_task(os_task_param_t task_init_data)
{
    /* Write your local variable definition here */

    /* Initialization of Processor Expert components (when some RTOS is active). DON'T REMOVE THIS CODE!!! */
#ifdef MainTask_PEX_RTOS_COMPONENTS_INIT
    PEX_components_init();
#endif
    /* End of Processor Expert components initialization. */

    #if (USAR_MUTEX==1)
        OSA_MutexCreate(&mutexrgb);
    #else
        OSA_SemaCreate(&semargb, 0);
    #endif

    ledrgb_init( );

#ifdef PEX_USE_RTOS
    while (1) {
#ifdef
        /* Write your code here ... */

        OSA_TimeDelay(10);          /* Example code (for task release) */

#ifdef PEX_USE_RTOS
    }
#endif
    }
}
```

6. Upload do LAB2 no GitHub: <https://github.com/Intosi/CE-235>