QXD0145 - Sistemas de Tempo-Real

Introdução ao FreeRTOS

André Ribeiro Braga

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

14/03/2023

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Multitasking em Sistemas Embarcados

Porque utilizar um *kernel* de tempo-real?

- Há técnicas para *software* embarcado sem *kernel*
- Apropriadas para sistemas simples
- Casos mais complexos ⇒ Preferível usar um *kernel*
- Regras para a definição do limiar são subjetivas
- Priorização de tarefas ⇒ Aplicação atende aos prazos especificados

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Multitasking em Sistemas Embarcados

Outras vantagens para utilizar um *kernel*

- Abstração da temporização
- Manutenção/Extensão
- Modularidade
- Desenvolvimento em equipes
- Testagem precoce
- Reuso de código
- Melhora na eficiência
- Tempo de ociosidade (*idle*)
- Gerenciamento de energia
- Flexibilidade com interrupções
- Requisitos de processamento mistos

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Sobre o FreeRTOS

- Desenvolvido e mantido pela *Real Time Engineers Ltd.*
- Ideal para sistemas de tempo-real em MCUs e MPUs
- Aplicações com um mix de requisitos soft e hard real-time
- *Kernel* de tempo-real no topo de aplicações embarcadas
  - Objetivo de atender restrições temporais impostas
- Organização da aplicação em *threads* independentes
- Atribuição de prioridades de acordo com a criticidade

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Características do FreeRTOS

- Preempção e cooperação
- Atribuição de prioridades flexível
- Notificação de *tasks*
- Filas
- Semáforos binários/contáveis
- Mutex (recursivos)
- Temporizadores de *software*
- Grupos de eventos
- Funções *hook*
- Monitoramento de pilha
- Coleta de dados de execução

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Variantes do FreeRTOS

- FreeRTOS
  - Pode ser utilizado em aplicações comerciais
  - Disponível sem custo
  - Usuários mantém propriedade intelectual
- OpenRTOS
  - Inclui suporte comercial da Amazon
- SafeRTOS
  - Mesmo modelo de utilização
  - Desenvolvido para submissão de processos de certificação

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Distribuição FreeRTOS

- FreeRTOS *Ports*
  - Combinação compilador + arquitetura ⇒ Um *port*
  - Aproximadamente 20 compiladores / mais de 30 arquiteturas
- O *build*
  - Biblioteca que provê capacidade de multiprogramação
  - Conjunto de código fonte em C (alguns comuns a todos os *ports*)
  - Arquivos fonte na compilação ⇒ FreeRTOS API disponível
  - Cada *port* oficial acompanha uma aplicação demo
    - Pré-configuada para compilar os arquivo fonte e cabeçalho corretos

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Distribuição FreeRTOS

- FreeRTOSConfig.h
  - Configuração através de um único arquivo de cabeçalho
  - Deve estar associado a uma aplicação específica
  - Toda aplicação demo contém seu cabeçalho
  - Sempre recomendável utilizar um cabeçalho pré-existente
- Único arquivo .zip
  - Pode ser baixado em https://www.freertos.org/
  - Todos os *ports* disponíveis / várias aplicações demo pré-configuradas

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Distribuição FreeRTOS

Árvore de diretórios

```
FreeRTOS
    ├─Source   Directory containing the FreeRTOS source files
    └─Demo     Directory containing pre-configured and port specific FreeRTOS demo projects
FreeRTOS-Plus
    ├─Source   Directory containing source code for some FreeRTOS+ ecosystem components
    └─Demo     Directory containing demo projects for FreeRTOS+ ecosystem components
```

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Distribuição FreeRTOS

Código fonte comum

- tasks.c
- list.c
- queue.c
- timers.c
- event_groups.c
- croutine.c

```
FreeRTOS
    └─Source
        ├─tasks.c          FreeRTOS source file - always required
        ├─list.c           FreeRTOS source file - always required
        ├─queue.c          FreeRTOS source file - nearly always required
        ├─timers.c         FreeRTOS source file - optional
        ├─event_groups.c   FreeRTOS source file - optional
        └─croutine.c       FreeRTOS source file - optional
```

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Distribuição FreeRTOS

Código fonte específico

- Diretório *portable*
- Hierarquia
  - Compilador
  - Arquitetura
- 5 opções de *heap*

```
FreeRTOS
  └─Source
    └─portable Directory containing all port specific source files
      ├─MemMang Directory containing the 5 alternative heap allocation source files
      ├─[compiler 1] Directory containing port files specific to compiler 1
      │ ├─[architecture 1] Contains files for the compiler 1 architecture 1 port
      │ ├─[architecture 2] Contains files for the compiler 1 architecture 2 port
      │ └─[architecture 3] Contains files for the compiler 1 architecture 3 port
      └─[compiler 2] Directory containing port files specific to compiler 2
        ├─[architecture 1] Contains files for the compiler 2 architecture 1 port
        ├─[architecture 2] Contains files for the compiler 2 architecture 2 port
        └─[etc.]
```

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Distribuição FreeRTOS

Inclusão de cabeçalhos

- Três diretórios devem ser fornecidos ao compilador
  - FreeRTOS/Source/included
  - FreeRTOS/Source/portable/[compilador]/[arquitetura]
  - Caminho para o FreeRTOSConfig.h
- O arquivo fonte que utilizar a API do FreeRTOS deve incluir
  - FreeRTOS.h
  - Cabeçalhos com declarações das funções a serem utilizadas

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Distribuição FreeRTOS

Aplicações Demo

- Cada *port* vem com pelo menos uma aplicação demo
  - Desenvolvido e testado em *host* Windows
- Objetivos
  - Fornecer exemplos de projetos pré-configurados
  - Proporcionar experimentação com mínimo *setup* ou conhecimento
  - Demonstração de como a API pode ser utilizada
  - Uma base de onde aplicações reais podem ser criadas

```
FreeRTOS
    └─Demo          Directory containing all the demo projects
        ├─[Demo x]  Contains the project file that builds demo 'x'
        ├─[Demo y]  Contains the project file that builds demo 'y'
        ├─[Demo z]  Contains the project file that builds demo 'z'
        └─Common    Contains files that are built by all the demo applications
```

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Distribuição FreeRTOS

Criando Projetos de um Demo

- Recomendável que sejam criados adaptando um dos demos
  - Garanta que o projeto demo compila e executa apropriadamente
  - Remova os arquivos que definem as *tasks* demo
    - Arquivos dentro da pasta Demo/Common
  - Delete todas as chamadas de função dentro da função main()
    - Menos prvSetupHardware() e vTaskStartScheduler()
  - Verifique se o projeto ainda compila

```
int main( void )
{
    /* Perform any hardware setup necessary. */
    prvSetupHardware();

    /* --- APPLICATION TASKS CAN BE CREATED HERE --- */

    /* Start the created tasks running. */
    vTaskStartScheduler();

    /* Execution will only reach here if there was insufficient heap to
    start the scheduler. */
    for( ;; );
    return 0;
}
```

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Distribuição FreeRTOS

Criando Projetos do Zero

- Utilizando um *tool chain*, crie um novo projeto
- Assegure que o novo projeto compile e execute no *target*
- Adicione os arquivos fonte do FreeRTOS
- Copie o cabeçalho FreeRTOSConfig.h adequado ao *port* desejado
- Adicione os seguintes caminhos ao projeto
  - FreeRTOS/Source/include
  - FreeRTOS/portable/[compilador]/[arquitetura]
  - Diretório que contém o cabeçalho FreeRTOSConfig.h
- Copie as configurações do compilador do projeto demo relevante
- Instale qualquer gerenciador de interrupção que seja necessário

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Distribuição FreeRTOS

Tipos de Dados e Convensões de Código

- Cada *port* tem um único cabeçalho portmacro.h
- Definições para dois tipos dependentes do *port*
  - TickType_t: contagem de tempo
  - BaseType_t: tipo de dado mais eficiente para a arquitetura
- Nomes de variáveis
  - Prefixadas por tipo
    - 'c' para char
    - 's' para int16_t
    - 'l' para int32_t
    - 'x' para BaseType_t e outras estruturas não-padrão
    - 'u' para *unsigned*
    - 'p' para ponteiro

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Distribuição FreeRTOS

Tipos de Dados e Convensões de Código

- Funções são prefixadas pelo tipo de retorno e o arquivo fonte
- Exemplos:
  - vTaskPrioritySet()
  - xQueueReceive()
  - pvTimerGetTimerID()
- Existem também funções privadas ('prv')
- Macros são escritas em letras maiúsculas
- Macros são prefixadas com a indicação de onde é definida
  - 'port', 'task', 'pd', 'config', 'err'