

1 Introdução

O objetivo do projeto final da disciplina Sistemas Operacionais é possibilitar ao estudante desenvolver uma aplicação prática utilizando um sistema operacional de tempo real. Neste projeto serão utilizadas as ferramentas de controle oferecida pelo FreeRTOS (www.freertos.org) para garantir a execução coordenada e a sincronização de duas *tasks*.

Na apresentação do projeto, o grupo deverá explicar o código e o projeto no ambiente de desenvolvimento *Code Composer Studio*. A solução apresentada por cada grupo deverá ser justificada; e uma demonstração do projeto funcionando na placa deve ser apresentada.

Além disso, para complementar a apresentação, cada equipe deverá montar um relatório de avaliação usando a ferramenta *Trace* (http://www.freertos.org/FreeRTOS-Plus/FreeRTOS_Plus_Trace/FreeRTOS_Plus_Trace.shtml) para o FreeRTOS.

2 Projeto Básico

O projeto é baseado no porte do FreeRTOS para a placa da TI Tiva C Series, que vem como exemplo do *Code Composer Studio* da própria TI e possui duas *tasks*, uma que monitora o pressionamento das chaves da placa (*switch*) e outra que aciona os LEDs da placa (*led*). Para o projeto as chaves não serão utilizadas.

2.1 Descrição

Neste projeto existirão 3 *tasks* (**Temperatura; Grava; Serial**), todas elas acessam a interface UART para enviar dados para o PC via comunicação serial/USB usando um terminal virtual (sugestão: *Putty*). Por se tratar de um ambiente multitarefa, cada *task* pode preeminir outra a qualquer momento. Então, para garantir que o acesso ao recurso compartilhado seja feito de forma segura e coerente, um *mutex* deve ser utilizado.

A aplicação funciona da seguinte forma: a *task* Temperatura lê periodicamente (a cada 1000 ms) o sensor de temperatura do processador Tiva C, calcula a média aritmética e envia para as *tasks* Grava e Serial por meio de 2 filas (*queues*).

A *task* Grava armazena os valores lidos da fila em um *Buffer* de 10 posições até que seja totalmente preenchido. Após o preenchimento do *Buffer*, os dados são transferidos para a memória Grava do microcontrolador e o *Buffer* é esvaziado, reiniciando seu processo de preenchimento.

A *task* Serial funciona de forma similar à *task* Grava. Ela lê os dados enviados pela *task* Temperatura por meio de outra fila e envia por meio da UART os 10 últimos valores lidos.

A comunicação e coordenação entre as tarefas são executadas por meio das duas filas (Temperatura \Rightarrow Grava e Temperatura \Rightarrow Serial) e do *mutex*.

A figura a seguir ilustra como a solução deve ser implementada.

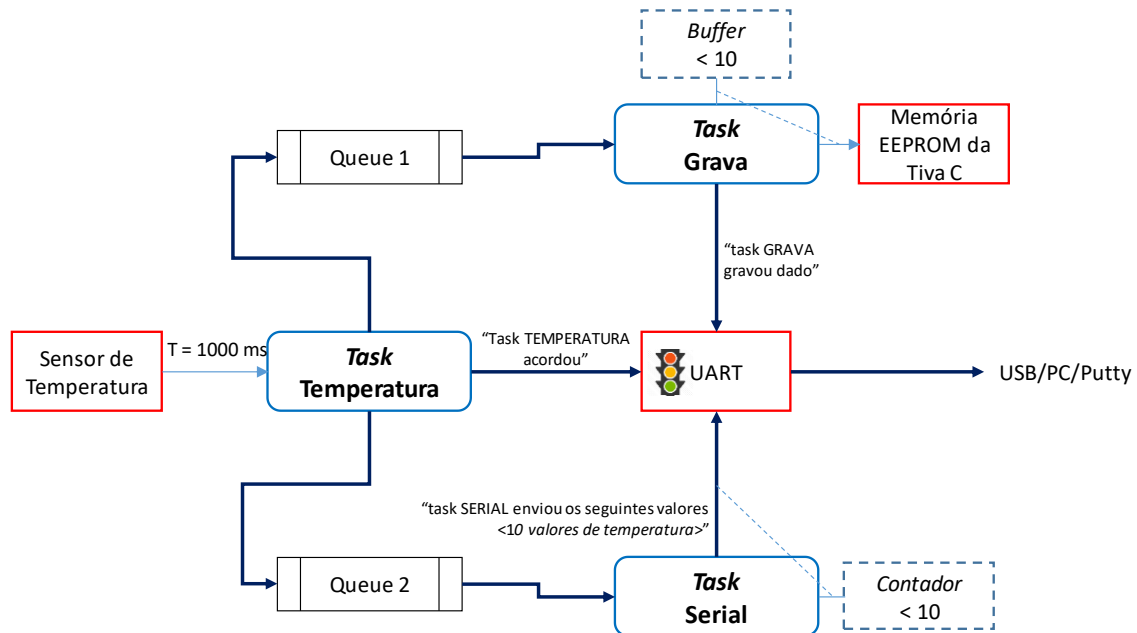


Figura 2.1 – Estrutura funcional do projeto.

Para indicar a visualização da *task* em execução, o LED RGB da placa de ser ativado com uma cor diferente sempre que uma *task* estiver executando:

- LED vermelho -> *task* Temperatura
- LED verde -> *task* Grava
- LED azul -> *task* Serial

Após a execução de cada *task*, o LED correspondente deve ser apagado.

2.2 Terminal Virtual

Assim como no porte de exemplo, o projeto deve se comunicar com um PC via porta serial virtual, usando a UART da placa Tiva C. A comunicação deve ser monitorada por um aplicativo de terminal virtual, como o *Putty*. O terminal virtual será “alimentado” por mensagens enviadas pelas *tasks*. Note, portanto, que o acesso à UART deve ser coordenado por um *mutex* do **FreeRTOS**. O terminal virtual no PC deve exibir mensagens das *tasks* de acordo com a seguinte descrição:

1. **Temperatura:** sempre que esta *task* “acordar” (lembre-se que ela executa periodicamente, ou seja, acorda para ler o sensor a cada 500 ms depois dorme) ela envia uma mensagem para o terminal virtual via UART com o conteúdo “Task TEMPERATURA acordou”;
2. **Grava:** sempre que um dado for armazenado na memória EEPROM, a mensagem “task GRAVA gravou dado” deve ser enviada para o PC;
3. **Serial:** a mensagem “task SERIAL enviou os seguintes valores <10 valores de temperatura>” deve ser enviada ao PC quando 10 valores de temperatura forem acumulados.

2.3 Sensor de Temperatura e LED

Um exemplo de projeto utilizando os recursos Sensor de Temperatura e LED RGB da Tiva C pode ser encontrado no *WorkBook* da placa: "TM4C123G_LaunchPad_Workshop_Workbook.pdf".

3 Informações complementares

- Modelo da placa
 - Tiva C series: <http://www.ti.com/tool/ek-tm4c123gxl>
- Ambiente de desenvolvimento:
 - *Code Composer Studio* (CCS): <http://www.ti.com/tool/ccstudio>
- Terminal virtual
 - Putty: <http://www.putty.org/>

4 Avaliação

Cada equipe deverá entregar o seguinte:

- Estrutura da solução no FreeRTOS (3,0 pts)
- Projeto funcional no ambiente CCS (2,5 pts)
- Análise com a ferramenta Tracyalyzer (2,0 pts)
- Explicação da solução (para o professor) (2,5 pts)