

Laboratório 3: Programação Concorrente com RTOS

Contexto:

No Lab 3 deve-se utilizar programação concorrente com RTOS, interrupções e periféricos integrados de um microcontrolador para implementar um sistema de controle.

Objetivo:

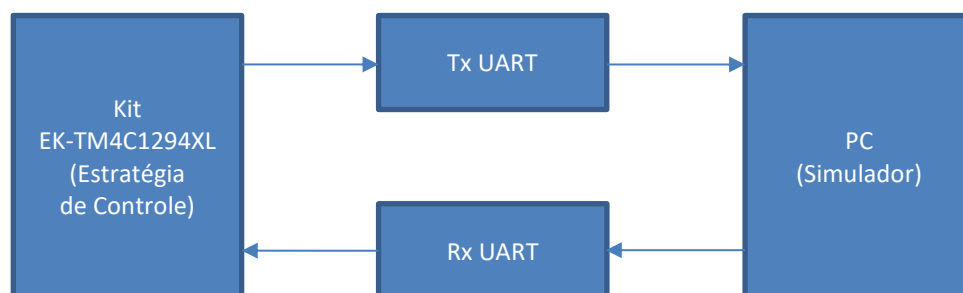
Escrever um programa concorrente em linguagem C (ou C + Assembly) utilizando o RTOS RTX5 para o kit EK-TM4C1294XL para uma das seguintes alternativas de projeto, a escolher:

1. Sistema com três elevadores em um prédio com 15 andares (simulador em PC)
2. Veículo autoguiado percorrendo uma pista oval com obstáculos (simulador em PC)
3. Controle de velocidade e sentido de rotação de um micromotor CC (hardware)

Qualquer das opções de projeto deverá ser implementada em um programa concorrente com pelo menos três tarefas. Comunicação serial (UART), quando necessária, deve ser implementada por interrupção, tanto na recepção quanto na transmissão de caracteres.

No caso das alternativas de projeto 1 e 2, a parte mecânica de cada sistema será simulada utilizando um dos simuladores desenvolvidos no projeto SimSE2 (ver links no website da disciplina). A interligação física entre o simulador (software rodando no PC) e o kit de desenvolvimento se dará via porta serial (COM virtual sobre USB). O simulador reage a comandos enviados pela porta serial do PC e informa o status do sistema de elevadores ou do veículo autoguiado também pela porta serial do PC. A documentação do protocolo de comunicação está inclusa no pacote de software do simulador.

O diagrama em blocos básico para as alternativas de projeto 1 ou 2 é:

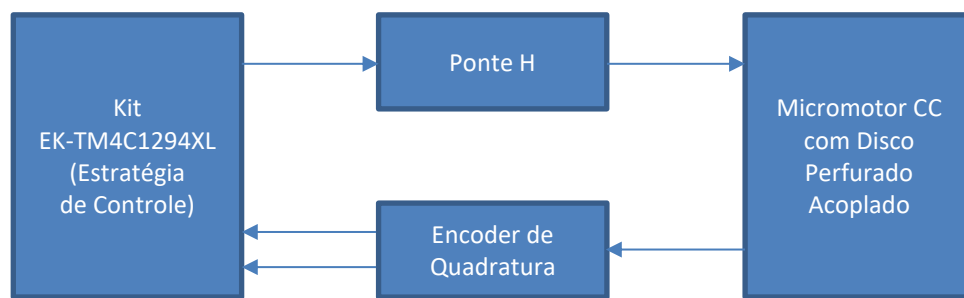


No caso da alternativa de projeto 3, poderá ser necessária a construção de hardware, uma vez que este envolve a necessidade de controle de velocidade e sentido de um micromotor CC físico. Poderá ser utilizada uma ponte H construída pela própria equipe ou algum kit desenvolvido por terceiros (por exemplo, o chamado “BoosterPack do Prof. Peron” possui circuitos adequados para acionamento de potência: um CI L293D e um CI ULN2003A). O professor irá disponibilizar a plataforma mecânica contendo o micromotor CC a ser controlado (AK360/53PL12S12500S¹) e um encoder de quadratura² implementado com duas chaves óticas e um disco perfurado acoplado ao eixo do micromotor.

¹ <https://neomotion.com.br/wp-content/uploads/2019/07/MMDC-R3-Sem-Red.rev01.pdf>

² <http://www.creative-robotics.com/quadrature-intro>

O diagrama em blocos básico para a alternativa de projeto 3 é:



Especificações do hardware da alternativa de projeto 3:

- O hardware utilizado pela equipe deve **necessariamente** estar montado de forma definitiva em placa de circuito impresso (universal ou específica) – **não** será permitida a conexão de montagens feitas provisoriamente em protoboard ao kit de desenvolvimento.
- O funcionamento do hardware utilizado pela equipe deve **necessariamente** ser testado **antes** da sua conexão com o kit, que deve **necessariamente** ser realizada com conectores **apropriados** (conector *header* macho ou conector *header* fêmea com *flat cable*).

O controle de velocidade e sentido de rotação do motor CC deve ser feito através de um ou mais sinais PWM gerados por hardware ou interrupções. A contagem de pulsos ou temporização do período entre pulsos gerados pelo encoder de quadratura para medição da velocidade e sentido de rotação do micromotor CC e realimentação da estratégia de controle também deve ser realizada por hardware ou interrupções (pode haver reaproveitamento de ideias do Lab 2).

Especificações do software da alternativa de projeto 3:

A funcionalidade esperada para o software é a de controlar a velocidade e o sentido de rotação do motor no *set-point* estabelecido pelo usuário, utilizando os dispositivos de entrada e saída do kit de desenvolvimento como interface homem-máquina (por exemplo, do BoosterPack Educacional ou do “BoosterPack do Prof. Peron”) ou UART + emulador de terminal.

O programa concorrente deve possuir ao menos três tarefas:

- Uma tarefa para a determinação do *set-point* de operação do motor (velocidade e sentido de rotação desejadas) utilizando os dispositivos de entrada disponíveis no kit (botões, *joystick* ou via comunicação serial, à escolha da equipe), além da visualização das informações correntes de operação em um display ou via comunicação serial, à escolha da equipe.
- Uma tarefa para contagem de pulsos ou temporização do período entre pulsos gerados pelo encoder de quadratura para medição da velocidade e sentido de rotação do micromotor.
- Uma tarefa para a implementação da estratégia de controle da velocidade e sentido de rotação à escolha da equipe. A estratégia de controle deverá procurar manter a velocidade de rotação do micromotor CC constante independentemente de perturbações externas.

Informações a serem mostradas no display (ou no software de terminal):

- Instruções sobre como utilizar os dispositivos de entrada do kit para alterar o *set-point*.
- Ponto de operação desejado (*set-point*), em RPM (valor sinalizado para indicar o sentido).
- Ponto de operação real (medido), em RPM (valor sinalizado para indicar o sentido).

Referências bibliográficas relevantes para a alternativa de projeto 3:

1. Joseph L. Jones, Anita M. Flynn, Bruce A. Seiger. Mobile Robots: Inspiration to Implementation, 2nd edition. A. K. Peters, 1999 (exemplar físico disponível na Biblioteca Central da UTFPR Curitiba).
2. Wescott, Tim. PID Without a PhD, Wescott Design Services, 2016 (arquivo em formato PDF disponível no *website* da disciplina).

Cronograma para todas as alternativas de projeto:

- **31/10/2018 (S11) e 01/11/2019 (S12)** – Planejamento da solução: levantamento de requisitos funcionais e não funcionais; esboço da arquitetura física do sistema. Testes dos simuladores e protocolos de comunicação com emulador de porta serial virtual e emulador de terminal de comunicação (apenas para as alternativas de projeto 1 ou 2).
- **07/11/2019 (S11) e 08/11/2019 (S12)** – Entrega parcial (projeto): projeto da arquitetura – diagrama de classes / objetos ativos do sistema de controle, projeto detalhado dos componentes – diagramas de atividades ou de estados de cada tarefa, diagrama esquemático do bloco de hardware (este último apenas para a alternativa de projeto 3).
- **14/11/2019 (S11) e 22/11/2019 (S12)** – Entrega parcial (código): por exemplo, funções de controle de periféricos ou rotinas de tratamento de interrupção.
- **28/11/2019 (S11) e 29/11/2019 (S12)** – Entrega parcial (código): por exemplo, funções das tarefas do sistema.
- **05/12/2019 (S11) e 06/12/2019 (S12)** – Apresentação do sistema completo funcionando (*observação: datas da segunda prova teórica*).