

Pojeto Final - 2024.1

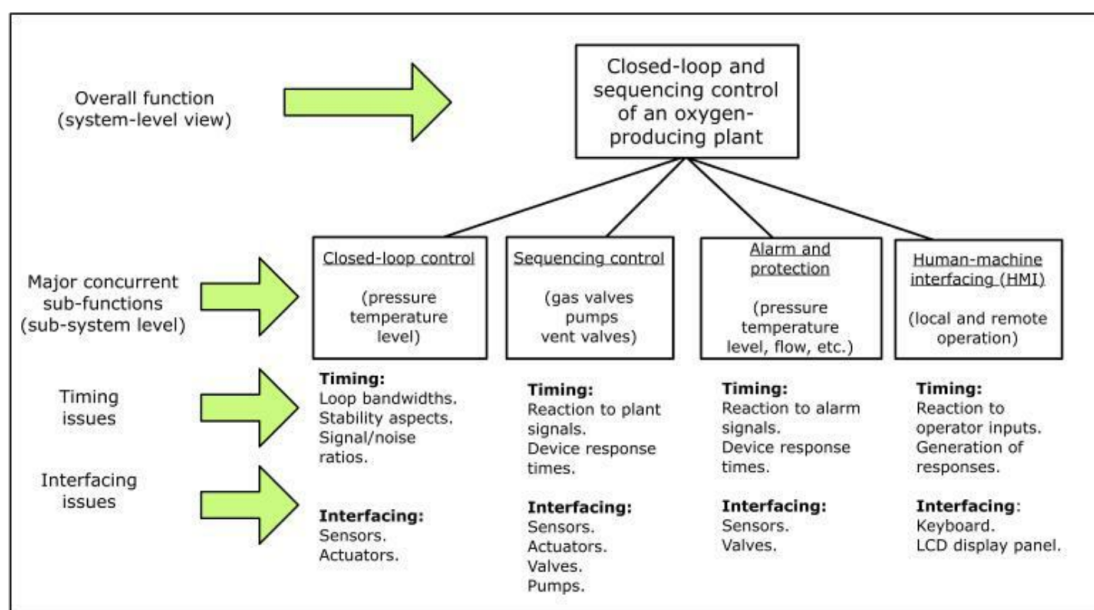
O objetivo deste projeto é reforçar os conhecimentos abordados ao longo da disciplina relacionados ao projeto de sistemas embarcados utilizando um sistema operacional de tempo real (RTOS), mais especificamente o **FreeRTOS** a partir da STM32CubeIDE [1, 2, 3, 4].

1 Instruções Gerais

O projeto deve ser feito utilizando um RTOS e utilizando as estruturas comuns desse tipo de projeto:

- **Projeto baseado em *tasks*** [5]: seguindo a metodologia resumida na Figura 1;

Figura 1: Resumo da metodologia de projeto baseada em *tasks*.



Fonte - [5].

- **Uso de Semáforos:** quando for necessário sincronismo unilateral e bilateral;
- **Uso de Mutexs:** quando for necessário o acesso de recursos compartilhados ;
- **Uso de Queues:** quando for necessário transferir dados entre *tasks*;

2 Projetos

Seguem ideias para dois possíveis projetos, cada aluno pode escolher qual fazer, desde acordado com o professor e que tenha material suficiente a disposição. Se quiser, o aluno pode

sugerir um projeto específico ou modificações na ideia apresentada neste documento. No entanto este projeto precisa ser aceito pelo professor.

2.1 Sugestões de Projetos

- **Relógio Digital:**

- Utilizando o display de 7 segmentos disponível no *Multifunction-Shield* Arduino¹, coloque para ficar aparecendo a hora e minutos;
- Use o ponto para mostrar os segundos passando (1s acesso, 1s apagado)
- Utilizando os três botões do *Multifunction-Shield* adicione as seguintes funcionalidades:
 - * **Atualizar a hora:** a partir do modo de mostrar as horas, uma vez que seja pressionado o botão SW1 o relógio entra no modo de atualização da hora, use os três botões para conseguir mudar o valor, mudar qual o dígito está sendo modificado e confirmar a nova hora. Use o dígito piscando para o usuário saber qual dígito está sendo atualizado e a buzina para indicar quando um valor é confirmado.
 - * **Cronômetro:** a partir do modo de mostrar as horas, uma vez que seja pressionado o botão SW2 o relógio entra no modo cronômetro. Use os botões para inicializar a contagem de tempo (agora o display deve conta décimos de segundos e segundos), pausar, reiniciar e voltar para o modo de mostrar as horas; use a buzina para indicar quando uma ação é tomada.
 - * **Temporizador:** a partir do modo de mostrar as horas, uma vez que seja pressionado o botão SW3 o relógio entra no modo temporizador. Use os botões para que o usuário marcar um tempo (minutos e segundos) que o temporizador irá contar de forma regressiva. Use a buzina da placa para gerar um alarme quando a contagem chegar a zero.

- **Medidor de temperatura Bluetooth:**

- A temperatura será medida periodicamente utilizando um sensor;
- A temperatura é para ser mostrada em tempo real no display de 7 segmentos disponível no *Multifunction-Shield* Arduino
- O valor da temperatura também deve ser enviado para um smartphone por Bluetooth. Para isso será utilizado o Módulo Bluetooth BLE V4.0 HM-10 Keyes².

3 Entrega

Será esperado que cada aluno entregue três coisas para dar como finalizado o projeto:

1. Mostrar o sistema funcionando em sala de aula;
2. Entregar o código utilizado no protótipo funcional;
3. Entregar o relatório com o descritivo do que foi feito;

¹Mais sobre: <https://blog.eletrogate.com/guia-completo-do-shield-multi-funcoes-para-arduino/>.

²Datasheet: <https://people.ece.cornell.edu/land/courses/ece4760/PIC32/uart/HM10/DSD%20TECH%20HM-10%20datasheet.pdf>.

4 Relatório

No relatório é esperado que seja mostrado a estrutura das *tasks* utilizadas em organização semelhante a mostrada na Figura 1.

Além disso, é esperado que seja descrito como configurar e utilizar cada um dos módulos periféricos que forem utilizados e que não tenham sido já utilizados nas práticas anteriores. Por exemplo, como configurar saídas PWM, como configurar Temporizadores, como configurar e utilizar o módulo Bluetooth, etc.

Para atingir o objetivo requerido no parágrafo anterior, utilizar estrutura semelhante a utilizada no descritivo das práticas da disciplina. Em que *screenshots* da tela da IDE são utilizadas para mostrar o passo-a-passo da configuração de periféricos. Assim como quais são as funções necessárias para interagir com um dado periférico e que valores de entrada e saídas são esperados. A descrição em detalhes das funções e bibliotecas se faz ainda mais necessária quando elas forem de autoria do aluno.

Referências

- [1] ST, *UM1722 User manual: Developing applications on STM32Cube with RTOS*, rev 3 ed., Oct 2019. Disponível em: https://www.st.com/resource/en/user_manual/dm00105262-developing-applications-on-stm32cube-with-rtos-stmicroelectronics.pdf.
- [2] FreeRTOS, “Api reference.” Disponível em: <https://www.freertos.org/a00106.html>, 2021. Acessado em 22/04/2021.
- [3] J. Cooling, *Real-time Operating Systems: Book 2 - The Practice (Using STM Cube, FreeRTOS and the STM32 Discovery Board)*. Lindentree Associates, 2018.
- [4] João Ranhel, *Apostila: Sistemas Microprocessados – ARM CORTEX*, ver 5 ed., 2019.
- [5] J. Cooling, *Real-time Operating Systems: Book 1 - The Theory (The engineering of real-time embedded systems)*. Lindentree Associates, 2018.