



Prof.: Guilherme de Santi Peron

LAB 1 - GPIO e Interrupções

Objetivo:

- Interfacear com entrada e saída genérica;
- Implementar uma interrupção de GPIO;
- Implementar um código para simular um sistema de malha fechada, em que há um setpoint de temperatura e o sistema deve chegar a esta temperatura desejada.
- Utilizar instruções Assembly para Cortex-M4 e o kit de desenvolvimento EK-TM4C1294-XL.

Contexto:

- Imaginar um termostato digital em que há uma variável contendo a temperatura atual e uma variável contendo o setpoint com a temperatura alvo. O sistema deve atuar para levar a variável de processo até o valor do setpoint.

Tarefas:

- Estudar a seção dos Displays de 7 Segmentos e LEDs da PAT DAELN;
- Fazer o fluxograma do código planejado conforme o roteiro;
- Implementar o código conforme o roteiro e o fluxograma utilizando instruções Assembly para Cortex-M4;
- **Mostrar para o professor e depois entregar a pasta do projeto Keil com todos os arquivos zipada, a imagem fluxograma (pdf, jpg ou png) da ideia proposta também dentro da pasta (preferencialmente em algum site ou aplicativo, e.g. <http://draw.io>). Nomear o arquivo com o nome e o último sobrenome dos alunos da equipe. Ex.: **fulanodetal1_fulanodetal2_fulanodetal3_ap1.zip**. Apenas um membro da equipe precisa enviar.**



Roteiro:

- 1) O sistema terá duas **variáveis de controle** principais:
 - a) Temperatura atual: um valor que simula a temperatura do ambiente. Ao ligar o sistema, ela deve inicializar em 15°. Para tornar o sistema dinâmico, a cada 1 segundo aproximadamente, essa variável deve oscilar, incrementando ou decrementando 1° conforme a temperatura alvo. Este valor deve ser exibido nos displays de 7 segmentos da PAT;
 - b) Temperatura alvo (setpoint): o valor desejado para a temperatura. Ao ligar, deve se inicializar em 25°. Este valor deve ser mostrado em BINÁRIO nos 8 LEDs da PAT.
- 2) O sistema deve comparar continuamente a temperatura atual e a temperatura alvo, para decidir qual ação tomar. A lógica deve ser mostrada nos LEDs da EK-TM4C1294XL:
 - a) Se a temperatura atual for maior que o setpoint, o LED ligado ao PN1 deve acender, indicando que o sistema está em resfriamento;
 - b) Se a temperatura atual for menor que o setpoint, o LED ligado ao PN0 deve acender, indicando que o sistema está em aquecimento;
 - c) Se ambas as temperaturas forem iguais, o sistema está em equilíbrio e ambos os LEDs (PN0 e PN1) devem acender.
- 3) Os botões **USR_SW1** e **USR_SW2** devem ser configurados para gerar uma interrupção por borda de descida:
 - a) USR_SW1: Ao ser pressionado, o valor da temperatura alvo (setpoint) deve ser incrementado em 1. O novo valor deve ser imediatamente refletido nos LEDs. O valor máximo é 50°.
 - b) USR_SW2: Ao ser pressionado, o valor da temperatura alvo (setpoint) deve ser decrementado em 1. O novo valor deve ser imediatamente refletido nos LEDs. O SP não pode ser menor que 5°.



- 4) Ao se utilizar a PAT DAELN (observar esquemático), notar que os dois *displays* de 7 segmentos e os LEDs estão multiplexados pelos pinos **PB4**, **PB5** e **PP5**, que estão ligados nas bases de três transistores. Estes transistores devem alternar funcionamento com temporizações adequadas para garantir uma visualização estável e sem cintilação. (Ver dicas abaixo).

Dicas:

- 1) Método para ativar os displays de 7 segmentos e os LEDs:
 - a) Os displays de 7 segmentos e os LEDs estão multiplexados nos pinos PA7:PA4 e PQ3:PQ0, desta forma, para a visualização dos mesmos, deve-se acender por um instante de tempo cada um, observar no esquemático da PAT DAELN.
 - b) Por exemplo, para acender o display DS1, deve-se colocar a informação desejada em PA7, PA6, PA5, PA4, PQ3, PQ2, PQ1, PQ0, depois ativar o Q2 por um instante de tempo, por exemplo 1ms e depois desativar Q2 por um instante de tempo para fornecer um tempo de guarda, por exemplo, por mais 1ms.
 - c) Assim, recomenda-se:
 - i) Colocar a informação da dezena em PA7:PA4 e PQ3:PQ0;
 - ii) Ativar o transistor Q2
 - iii) Esperar 1ms
 - iv) Desativar o transistor Q2
 - v) Esperar 1ms
 - vi) Colocar a informação da unidade em PA7:PA4 e PQ3:PQ0;
 - vii) Ativar o transistor Q1
 - viii) Esperar 1ms
 - ix) Desativar o transistor Q1
 - x) Esperar 1ms
 - xi) Colocar a informação dos LEDs em PA7:PA4 e PQ3:PQ0;
 - xii) Ativar o transistor Q3



- xiii) Esperar 1ms
- xiv) Desativar o transistor Q3
- xv) Esperar 1ms

d) Para criar um temporizador de 1 segundo, usar o atraso total que você já implementou no item (c) como base. Primeiro, calcular quantas vezes N esse atraso precisa ser repetido para somar aproximadamente um segundo. Em seguida, criar uma variável contadora inicializada com N. A cada ciclo, executar o atraso do item (c) e decrementar o contador. A temperatura só deve ser alterada (incrementada ou decrementada) quando o contador chegar a zero. Ao chegar a zero, reiniciar o contador com o valor N para começar a contagem do próximo segundo.

- 2) Para escrever o valor da temperatura atual nos *displays* de 7 segmentos, pode-se utilizar as instruções **UDIV** por 10 para encontrar a dezena e **MLS** para encontrar o resto da divisão por 10 para a unidade.
- 3) Ambas as chaves USR1_SW1 e USR_SW2 entram na mesma rotina de tratamento de interrupção `GPIOPortJ_Handler`. Neste caso, para identificar qual das duas chaves foi pressionada e gerou a interrupção, deve ser realizada a leitura do `GPIORIS` ou `GPIOMIS` e realizada uma comparação e teste: se o bit 0 estiver ativado, a PJ0 gerou a interrupção; se o bit 1 estiver ativado, a PJ1 gerou a interrupção.