



Lista de Exercícios #4

Pado Labs - Microcontroladores

Analog-to-Digital Converter (ADC)

Tips and Tricks : Utilizar o *User Manual UM2324* e o *User Manual UM2319* para resolver as questões.

Requirements : Resolva pelo menos **6 exercícios**. Exercícios com a tag **Challenge** valem por dois exercícios.

Requirements : Exercícios que requerem desenvolvimento de um código devem ser enviados em repositórios no *Github*.

1: Um determinado projeto de refrigerador utiliza um sensor de temperatura condicionado a trabalhar de -75°C até 5°C (0V à VDD), este refrigerador implementa um PID que é alimentado por este sensor de temperatura. Como requisito, a precisão da temperatura medida deve ser de $0,02^{\circ}\text{C}$. Dado a condição, calcule a resolução mínima do ADC que atenda ao requisito do projeto.

2: Implemente um programa que leia um potenciômetro e converta o valor lido do ADC em tensão. (Lembrando que a alimentação do microcontrolador é de 3.3V)

3: Implementar um firmware que faça a leitura de um potenciômetro e, utilizando três LEDs, acenda os mesmos de acordo com a seguinte regra definida pela tabela 1:

Tabela 1: LEDs a serem acionados de acordo com cada tensão lida pelo ADC.

Valor do ADC Acionar

Até 1V LED1

Maior que 1V e Menor que 3V LED2

Maior que 3V LED3

4: Habilite dois canais do ADC e conecte a um mesmo potenciômetro. No primeiro canal, realize a leitura normalmente atribuindo a uma variável. No segundo canal, realize a média aritmética de 10 leituras e armezene em uma variável e compare o resultado obtido. Recomenda-se que o leitura do primeiro canal seja feita a cada 10 ciclos, para manter o sincronismo entre o canal em que é realizada a filtragem por média aritmética.

5: Faça um firmware que leia 4 potênciômetros utilizando 4 canais do ADC.

Associe 1

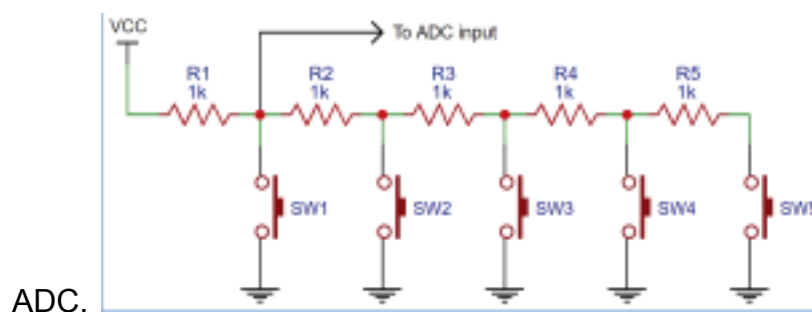
a estes potênciômetros quatro LEDs, onde o LED 1 é controlado pelo potênciômetro 1, e assim por diante. Quando a leitura do potênciômetro n for maior que 2,7V, ligar o LED n , caso contrário, deve manter o LED apagado. Neste exercício utilize a leitura dos canais por **polling**.

6: Repita o exercício anterior, mas neste contexto, utilize a leitura dos canais com o controlador *DMA* com conversão contínua habilitada.

7: Crie um programa que realize a leitura do sensor de temperatura interno do STM32G0B1RE. Converta o valor lido em °C.

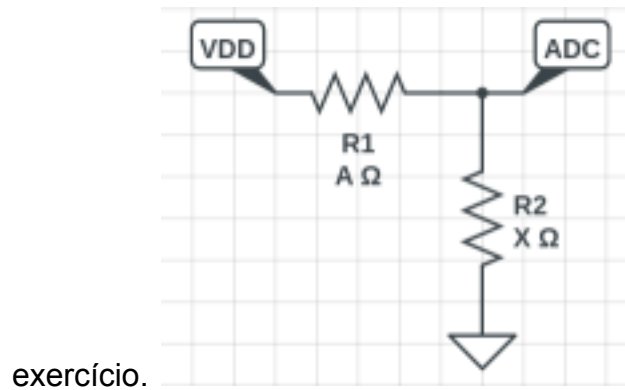
8: **Challenge:** Quando precisamos de botões em uma aplicação e utilizamos as GPIOs de forma convencional, em geral se utiliza uma para cada chave, ou um pouco menos quando se utiliza ligação em matriz. No entanto, com o uso do periférico de leitura analógica, é possível utilizar apenas uma entrada para ler vários botões. Disto, então, monte o circuito na figura 1 e implemente um firmware que faça a leitura das teclas utilizando o periférico de ADC. Obs: não é necessário utilizar os mesmos valores de resistores, montando pelo menos 4 botões.

Figura 1: Esquemático base para leitura de botões utilizando o



9: **Challenge:** Utilizando divisor de tensão, faça um programa que, utilizando leitura analógica, meça a resistência de um resistor. É possível determinar a resistência a partir da tensão que é lida com a equação 1, tendo como base o circuito da figura 2, onde R1 é um resistor fixo e R2 o resistor que terá sua resistência calculada.

Figura 2: Divisor de tensão para utilizar como base para elaboração do



$$R_2 = V_o$$

$$V_{DD} - V_o * R_1 (1)$$

10: Challenge: Implemente a leitura de um modulo *joystick*, armazene em uma *struct* os valores de X e Y, o valor tem de ir de -100 à 100, representando os extremos, e o valor 0 indica o ponto central. Na figura 3 é ilustrado o esquemático do módulo *joystick*.

Figura 3: Esquemático do KY-023.

