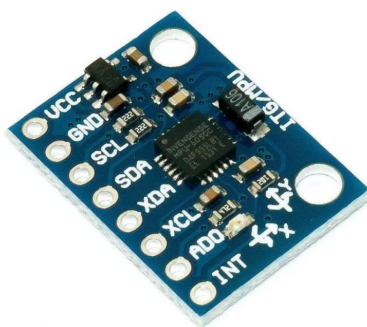


Exercícios referentes ao Capítulo 8: Tipos Abstratos de Dados

1. Exercícios de programação:

- (a) Um sensor MPU-6050 (ver figura abaixo) é um sensor de movimento inercial utilizado em projetos para Arduino e outros microcontroladores (é amplamente utilizado em projetos de robótica, drones, estabilização de câmeras, entre outras aplicações que necessitam de detecção de movimento em várias dimensões). Ele contém um acelerômetro e um giroscópio, cada um deles fornecendo medidas em 3 eixos no espaço, ou seja: ele consegue fornecer a aceleração nos eixos X, Y e Z (através do acelerômetro), e também consegue fornecer a velocidade angular (rotação) nos eixos X, Y e Z (através do giroscópio).

Figura 1: Sensor MPU-6050



A medida de aceleração que podemos calcular a partir dos dados fornecidos pelo sensor é expressa em g (gravidade terrestre), onde $1 g$ equivale aproximadamente a $9,81 \text{ m s}^{-2}$. Por exemplo, quando o sensor está em repouso sobre uma superfície plana e sem movimento, o acelerômetro normalmente medirá cerca de $1 g$ no eixo Z (se estiver alinhado com a gravidade), indicando que a aceleração da gravidade está puxando o sensor para baixo. Se o sensor estiver inclinado, os valores de aceleração serão distribuídos entre os eixos X, Y, e Z, de acordo com sua orientação.

O MPU-6050 tem quatro opções de configuração de sensibilidade para o acelerômetro, que afetam a escala dos valores medidos: $\pm 2 g$, $\pm 4 g$, $\pm 8 g$ e $\pm 16 g$. Essas configurações determinam o intervalo máximo de aceleração que pode ser medido para cada eixo, por exemplo:

- Na configuração de $\pm 2 g$ o sensor pode medir acelerações entre $-2 g$ a $2 g$, o que significa um intervalo de $-19,62 \text{ m s}^{-2}$ até $19,62 \text{ m s}^{-2}$; e
- Na configuração de $\pm 16 g$, o intervalo vai de $-16 g$ a $16 g$, ou seja, de $-156,96 \text{ m s}^{-2}$ até $156,96 \text{ m s}^{-2}$.

A medida da rotação (velocidade angular) que podemos calcular a partir dos dados fornecidos pelo sensor é expressa em $^\circ \text{ s}^{-1}$ (graus por segundo). Se o sensor estiver completamente em repouso (sem qualquer movimento ou rotação), em teoria, não deveria haver nenhuma rotação registrada pelo giroscópio, ou seja, os valores de rotação para os eixos X, Y e Z deveriam ser próximos de zero¹.

¹Na prática pode haver uma mínima quantidade de rotação registrada devido à ruídos na leitura do giroscópio e/ou de um fenômeno chamado de *drift*. Vamos ignorar a ocorrência dessas situações neste trabalho.

Assim como o acelerômetro, o giroscópio do MPU-6050 também tem quatro opções de configuração para a sensibilidade do giroscópio, que afetam o intervalo de medição: $\pm 250^\circ \text{ s}^{-1}$, $\pm 500^\circ \text{ s}^{-1}$, $\pm 1000^\circ \text{ s}^{-1}$ e $\pm 2000^\circ \text{ s}^{-1}$. Essas configurações definem o intervalo máximo de rotação (em graus por segundo) que o giroscópio pode medir em cada eixo.

Quando o MPU-6050 faz as medições da aceleração e da rotação ele **não** registra os valores finais nas unidades “corretas”. O que ele registra são seis números inteiros *signed* de 16 bits (que variam de $-32\,768$ até $32\,767$):

- ax: valor bruto da aceleração no eixo X;
- ay: valor bruto da aceleração no eixo Y;
- az: valor bruto da aceleração no eixo Z;
- rx: valor bruto da rotação no eixo X;
- ry: valor bruto da rotação no eixo Y; e
- rz: valor bruto da rotação no eixo Z.

A partir desses valores brutos registrados é possível calcular os valores corretos e finais para a aceleração e a rotação, dependendo de um **fator de escala** específico que é dado de acordo com as configurações de sensibilidade da aceleração e da rotação. O cálculo da aceleração é dado por:

$$\text{aceleração } g = \frac{\text{valor bruto da aceleração no eixo}}{\text{fator de escala da sensibilidade da aceleração}} \quad (1)$$

O cálculo da rotação é dado por:

$$\text{rotação } ^\circ \text{ s}^{-1} = \frac{\text{valor bruto da rotação no eixo}}{\text{fator de escala da sensibilidade da rotação}} \quad (2)$$

As tabelas abaixo mostram os fatores de escala para as diferentes configurações de sensibilidade da aceleração e da rotação:

Tabela 1: Fator de Escala para as Sensibilidades da Aceleração

| Sensibilidade da Aceleração | Fator de Escala |
|-----------------------------|-----------------|
| $\pm 2 g$ | 16 384 |
| $\pm 4 g$ | 8192 |
| $\pm 8 g$ | 4096 |
| $\pm 16 g$ | 2048 |

Tabela 2: Fator de Escala para as Sensibilidades da Rotação

| Sensibilidade da Rotação | Fator de Escala |
|---------------------------------|-----------------|
| $\pm 250^\circ \text{ s}^{-1}$ | 131,0 |
| $\pm 500^\circ \text{ s}^{-1}$ | 65,5 |
| $\pm 1000^\circ \text{ s}^{-1}$ | 32,8 |
| $\pm 2000^\circ \text{ s}^{-1}$ | 16,4 |

Agora que você sabe como funciona um sensor MPU-6050, sua tarefa é **criar um tipo abstrato de dado** (mpuTAD), que seja capaz de armazenar as seguintes informações:

- A configuração da sensibilidade da aceleração;
- Os valores brutos da aceleração para os três eixos;
- Os valores finais e corretos da aceleração para os três eixos;
- A configuração da sensibilidade da rotação;
- Os valores brutos da rotação para os três eixos; e
- Os valores finais e corretos da rotação para os três eixos.

Você deve especificar o tipo abstrato em uma interface apropriada (`mpuTAD.h`), com todos os comportamentos necessários para o correto e completo uso desse sensor, tais como: criar um tipo de dado MPU; inserir as leituras brutas dos sensores, recuperar as leituras brutas dos sensores, inserir a configuração de sensibilidade da aceleração e da rotação, recuperar as configurações de sensibilidade, recuperar os valores finais e corretos da aceleração e rotação, etc. Note que os valores finais e corretos da aceleração e rotação não devem ser inseridos pelo usuário: o usuário informará a configuração de sensibilidade e os valores brutos, e o tipo abstrato deverá calcular automaticamente os valores finais e corretos para cada eixo.

Depois que você especificar tudo o que for necessário na interface, crie uma implementação para a interface (`mpuTAD.c`). A representação interna do tipo abstrato e os subprogramas devem ser capazes de fornecer todos os comportamentos especificados na interface.

Crie também um pequeno programa cliente (`teste_mpu.c`) que utilize a interface e demonstre que o TAD e sua implementação estão corretos.

Quanto terminar, crie um arquivo comprimido (no formato zip) contendo a interface, a implementação da interface e o cliente de teste. Esse arquivo deverá ser enviado pelo portal até a data limite especificada.

- (b) O tipo abstrato de dado que você criou na primeira parte deste exercício (`mpuTAD.h`) é perfeito para armazenar os dados de apenas um único sensor. Mas pode ser necessário, em uma aplicação específica, armazenar dados de diversos sensores em uma **pilha** (o último sensor que entrou na pilha será o primeiro sensor a sair da pilha).

Sua tarefa é **criar uma pilha** para o armazenamento de até 100 sensores MPU. Essa pilha deve ser uma interface (`pilhampuTAD.h`) que especifica os comportamentos da pilha. Note que a pilha não precisa ser genérica, ela pode ser especificamente criada para armazenar somente os sensores MPU. Crie uma implementação adequada (`pilhampuTAD.c`) e também um programa cliente para testes (`teste_pilhampu.c`).

Ao terminar coloque esses três arquivos em um arquivo compactado (no formato zip) e envie no portal até a data limite especificada.