



UNIVERSIDADE FEDERAL DE RORAIMA
PRÓ-REITORIA DE ENSINO E EXTENSÃO
CENTRO DE CIÊNCIA E TECNOLOGIA
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
INTRODUÇÃO A SISTEMAS EMBARCADOS
PROFESSOR DR. HERBERT ROCHA

ANNY WALKER

BÁRBARA ZAMPERE

SISTEMAS EMBARCADOS: GYRO_CONTROLLER

1. Introdução

O prof. dr. Herbert Rocha, ministrante da disciplina de Introdução a Sistemas Embarcador (ISE), requereu que para finalização da disciplina fosse implementado um projeto de ISE com uso de microcontrolador, cujo objetivo ficaria a cargo dos alunos. Por esse motivo as autoras desse trabalho elaboraram uma ideia de sistema embarcado que em formato de controle para movimentar um objeto 3D.

2. Objetivo

O objetivo é implementar um sistema embarcado com uso de microcontrolador e sensor giroscópio e acelerômetro que sirva como controle que pode ser usado em um ambiente 3D para controlar objetos.

3. Estratégia

A estratégia é estabelecer a comunicação entre o microcontrolador e o ambiente 3D e transmitir dados do sensor giroscópio/acelerômetro, que aos serem recebidos serão tratados e aplicados na posição e rotação de um objeto 3D.

4. Ferramentas

Como microcontrolador foi escolhido o Arduino Uno, devido sua portabilidade, baixo custo e fácil acesso, e para sensor giroscópio/acelerômetros optamos pelo modelo MPU-6050, pela disponibilidade em estoque da UFRR. Como ambiente 3D/engine foi definido o Unity, por sua gratuidade e familiaridade das autoras com o ambiente de desenvolvimento e linguagem C#.

5. Implementação

O primeiro passo foi estabelecer a comunicação entre o Arduino e o Unity, para tal foi usada a porta serial COM3, que transmite e recebe dados. Os dados do MPU-6050 são transmitidos através das portas SDA e SCL, e uma das bibliotecas que oferecem métodos de leitura e escrita e atributos para capturar as mensagens dessas portas é a Wire.h, que utiliza um buffer para armazenar temporariamente os dados, que podem ser no máximo de 32 bytes.

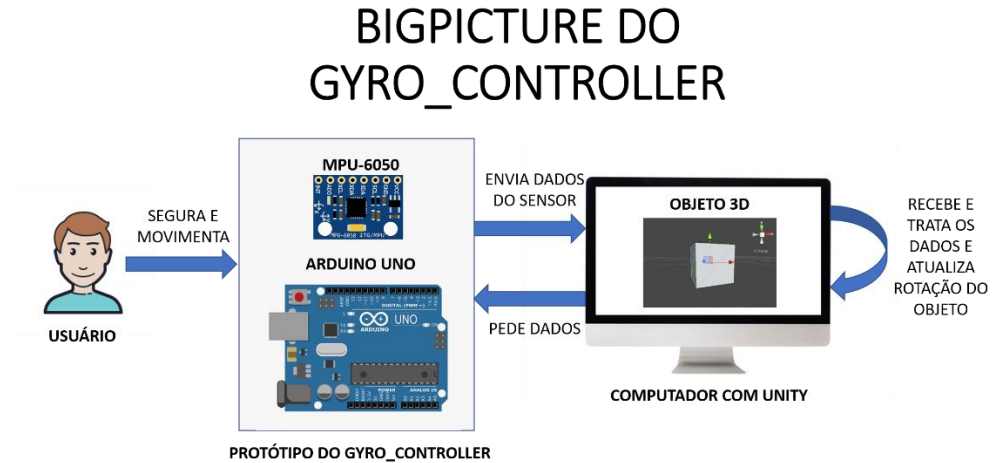
Conseguidos os dados e transmissão dos mesmo pela porta serial COM3, agora precisamos estabelecer a comunicação e recepção dos dados com o Unity, em um sistema operacional Windows. A partir do .NET 2.0 da Microsoft, obtivemos acesso ao pacote System.IO.Ports, compatível com C#, criamos um canal com a porta COM3 usando o objeto da classe SerialPort.

Essa classe oferece métodos de leitura e escrita usando a porta serial como canal, e é assim que obtemos no Unity os dados transmitidos pelo MPU-6050 conectado ao Arduino. A partir disso, os dados foram tratados de forma a serem compatíveis com ângulo de Euler, atributo dos objetos 3D do Unity, onde é possível inserir um vetor 3D rotacionado e assim atualiza o objeto virtual, que passar a corresponder aproximadamente a rotação do protótipo físico do controle.

6. BigPicture

O BigPicture da figura 1 abaixo, representa uma visão geral do uso e funcionamento do sistema.

Figura 1. BigPicture do Gyro_Controller



Fonte: Elaborado pelas autoras.

7. Storyboard

O Storyboard apresentado na figura 2 ilustra um caso de uso, onde uma cliente do Gyro_Controller possui o produto e como forma de sair do tédio utiliza o controle para jogar no Unity.

Figura 2. Storyboard do Gyro_Controller.



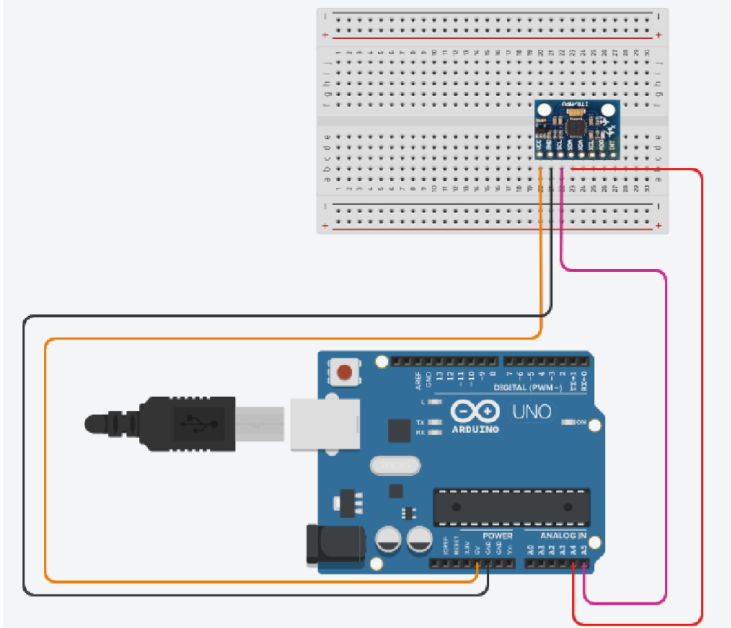
Fonte: Elaborado pelas autoras.

8. Prototipagem

A prototipagem do MPU-6050 é padrão e está demonstrada na figura 3 abaixo. As portas que transmitem os dados de rotação do MPU-6050 são SDA e SCL, que

respectivamente foram ligadas nas portas A4 e A5 do Arduino. O VCC e GND do sensor foram conectados ao 5V e GND do microcontrolador para permitir a condução de energia.

Figura 3. Prototipagem



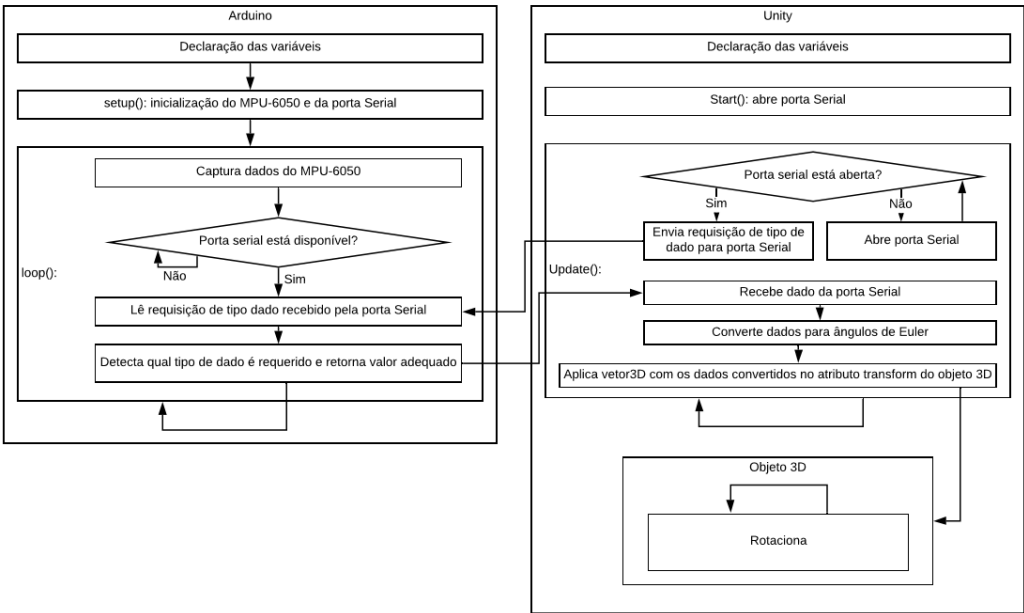
Fonte: Elaborado pelas autoras.

9. Fluxograma de execução

O fluxograma de execução da figura 4 mostra a ordem em que acontecem os eventos e qual a comunicação entre eles.

Figura 4. Fluxograma de execução.

FLUXOGRAMA DE EXECUÇÃO DO GYRO_CONTROLLER



Fonte: Elaborado pelas autoras.

10.Considerações finais

Por fim, após testes constatamos que existe um ruído na transmissão dos dados, onde mesmo o sensor parado estavelmente gera movimentação na rotação do objeto 3D, por falta de equipamento como medidor de calibrarem não conseguimos detectar a origem do ruído e qual seria o tratamento para tal problema. Mas o protótipo serve ao seu propósito, mesmo que de forma aproximada.

Fica a ressalva com relação a ideia inicialmente sugerida, em que o proposto era adicionar um botão a plataforma para que além de rotacionar o objeto, fosse possível também movimentar o objeto no espaço 3D. A barreira que encontramos foi a transmissão dos dados do botão do Arduino para o Unity através da porta serial com o uso da biblioteca Wire.h, que possibilita buffer de 32 bytes, apesar de sabermos que isso é suficiente para conter dados simples, não conseguimos encontrar um tipo de dado adequado que pudesse ser transmitido juntamente com os dados do MPU 60-50.