

Acadêmica: Gabriele Jennrich
Orientador: Miguel Alexandre Wisintainer

PROTÓTIPO DE MOUSE UTILIZANDO ACELERÔMETROS

Sumário

- ◉ Introdução
- ◉ Objetivos
- ◉ Fundamentação Teórica
 - Trabalhos correlatos
 - Desenvolvimento
 - Implementação
- ◉ Resultados
- ◉ Conclusões e Extensões

Introdução

Objetivos

- Desenvolver um protótipo de dispositivo para apontamento (*mouse*) que utilize os componentes eletrônicos acelerômetro e o microcontrolador PIC, como principal interface de interação entre o usuário e o sistema

Objetivos

- Traduzir os movimentos obtidos pelo acelerômetro
- Adequar as tradicionais funções de clique para compatibilizá-las com as limitações motoras do usuário e as do acelerômetro;
- Identificar um tipo de deficiência motora que possua características compatíveis com as do *kit* SEN-00410;
- Permitir ajustes no módulo de funcionamento , de forma que possa se adaptar as necessidades ou desejos do usuário.

Fundamentação Teórica

- Acelerômetro
- SEN-00410
 - Acelerômetro encapsulado
 - Tri-axial
 - PIC16LF88
 - Calibrável

```
COM1 - PuTTY
X=-1.118 Y=-2.938 Z= 0.253
X=-1.086 Y=-2.989 Z= 0.238
X=-1.075 Y=-2.979 Z= 0.365
X=-1.096 Y=-3.051 Z= 0.349
X=-1.118 Y=-2.979 Z= 0.301
X=-1.096 Y=-2.948 Z= 0.301
X=-1.129 Y=-2.938 Z= 0.301
X=-1.064 Y=-2.938 Z= 0.285
X=-1.086 Y=-2.979 Z= 0.238
X=-1.075 Y=-3.020 Z= 0.253
X=-1.118 Y=-2.969 Z= 0.349
X=-1.161 Y=-2.979 Z= 0.238
X=-1.075 Y=-2.989 Z= 0.285
X=-1.075 Y=-2.979 Z= 0.317
X=-1.043 Y=-2.928 Z= 0.349
X=-1.086 Y=-2.979 Z= 0.301
X=-1.086 Y=-2.938 Z= 0.301
X=-1.096 Y=-2.948 Z= 0.301
X=-1.118 Y=-2.979 Z= 0.301
X=-1.096 Y=-2.959 Z= 0.301
X=-1.118 Y=-2.948 Z= 0.238
X=-1.118 Y=-2.969 Z= 0.238
X=-1.075 Y=-2.938 Z= 0.238
X=-1.075 Y=-2.938 Z= 0.301
X=-1.075 Y=-3.020 Z= 0.301
```

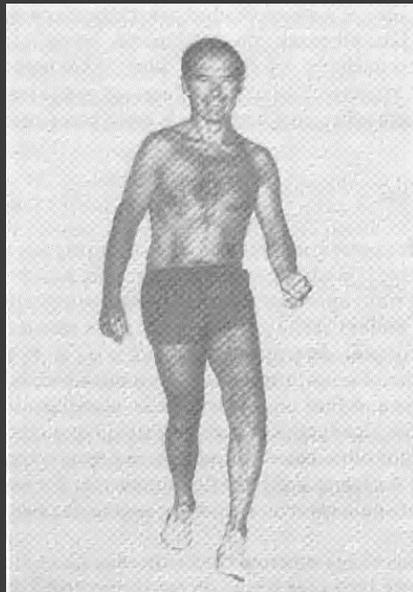
Fundamentação Teórica

- Grau de deficiência motora
 - Deficiência motora é a disfunção, congênita ou adquirida, dos membros superiores ou inferiores que acarrete em dificuldade de locomoção, coordenação motora ou fala.
 - Classificada pela quantidade de membros afetados e grau de comprometimento

Fundamentação Teórica

● Hemiplegia espástica

- A escápula está retraída e a cintura escapular em depressão (ombro caído);
- O punho está flexionado com algum desvio ulnar (punho dobrado apontando para baixo ou para cima);
- Os dedos estão fletidos e aduzidos (mão fechada);
- O tronco está rodado para trás no lado hemiplégico com flexão lateral do mesmo lado.



Trabalhos Correlatos

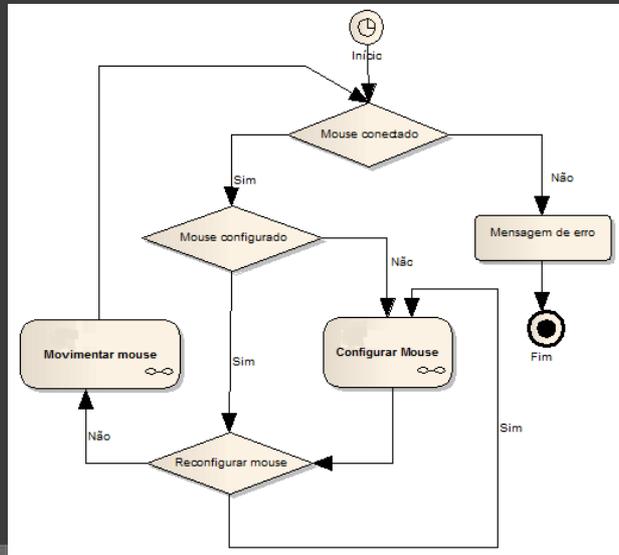
- Mouse com botões
- Mouse por toque
- Captura de movimentos
- Óculos mouse
- Nintendo Wii



Requisitos Funcionais

- Traduzir a inclinação do acelerômetro em movimentos no cursor do *mouse* (RF);
- Interpretar movimentos específicos que representam os atuais cliques de botão direito e esquerdo e comando de seleção(RF);
- Possuir uma interfase para configuração do *mouse*, onde o usuário poderá calibrar a inclinação para realização de cada um dos cliques e calibrar uma faixa de valores para a qual não exista movimentação (RF);
- Ser implementado utilizando o ambiente de programação Eclipse Galileo e Netbeans (RNF);
- Ser compatível com os sistemas operacionais Windows XP, Vista e Seven ou Linux (RNF).

Fluxograma Principal



Fluxograma da Movimentação

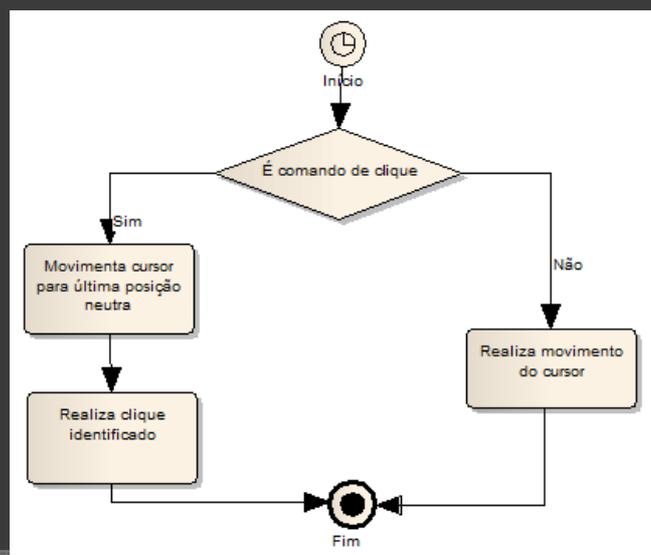
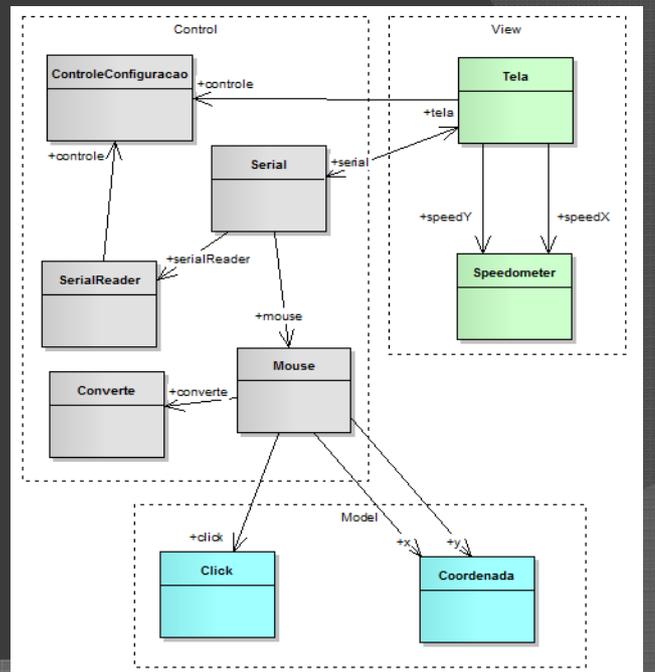


Diagrama de Classe



Implementação

- Linguagem de programação Java
- Biblioteca RXTX
- Biblioteca AWT (*Abstract Window Toolkit*)
- Dividido em 3 etapas:
 - Comunicação serial
 - Manipulação do cursor
 - Interface gráfica

Etapa 1: Comunicação

- Encontrar uma biblioteca capaz de comunicar-se com a serial através de um conversor;
- Percorrer todas as portas seriais;
- Criar uma thread para leituras da portas;
- Identificar a porta que atenda o padrão esperado;

Etapa 1: Comunicação – thread

```

1. connect(String portName) throws Exception {
2.     CommPortIdentifier portIdentifier =
CommPortIdentifier.getPortIdentifier(portName);
3.     (...)
4.     (...)
5.     (...)
6.     CommPort commPort = portIdentifier.open( ...
);
7.     if (!(commPort instanceof SerialPort)) {
8.         return false;
9.     }
10.    SerialPort serialPort = (SerialPort) commPort;
        serialPort.setSerialPortParams(...);
11.    InputStream in =
serialPort.getInputStream();
12.    serialReader = new SerialReader(in);
13.    Thread thread = new Thread(serialReader);
14.    thread.start();
15. thread.sleep(1000);
16.    (....)

```

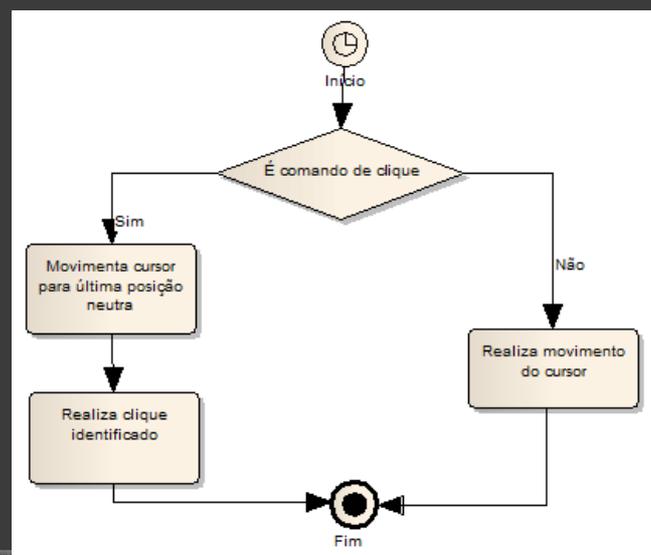
Etapa 1: Comunicação – busca do padrão

```

1. public void run() {
    (...)
6.     while ((len = this.in.read(buffer)) > -1) {
7.         le = (new String(buffer, 0, len));
8.         if (!le.contains("X=")) {
9.             ehPorta = false;
10.            continue;
11.        }
12.        ControleConfiguracao.controle(mouse, le);
13.        ehPorta = true;
    (...)
22. }

```

Etapa 2: Manipulação do cursor



Etapa 2: Manipulação do cursor – Realização do clique

```

1. void realizaClick(Double[] medias) {
2.     if (medias[0] <= click.clickSimples) {
3.         //valor representa clique simples
4.         if (!mouseClick) {
5.             // move para posição do clique
6.             mouseMoveClick();
7.             //clique simples
8.
9.             mousePress(InputEvent.BUTTON1_MASK);
10.            mouseDelay(10);
11.
12.            mouseRealise(InputEvent.BUTTON1_MASK);
13.            mouseClick = true;
14.        }
15.    }else (...)
```

Etapa 2: Manipulação do cursor – Movimentação nas coordenadas

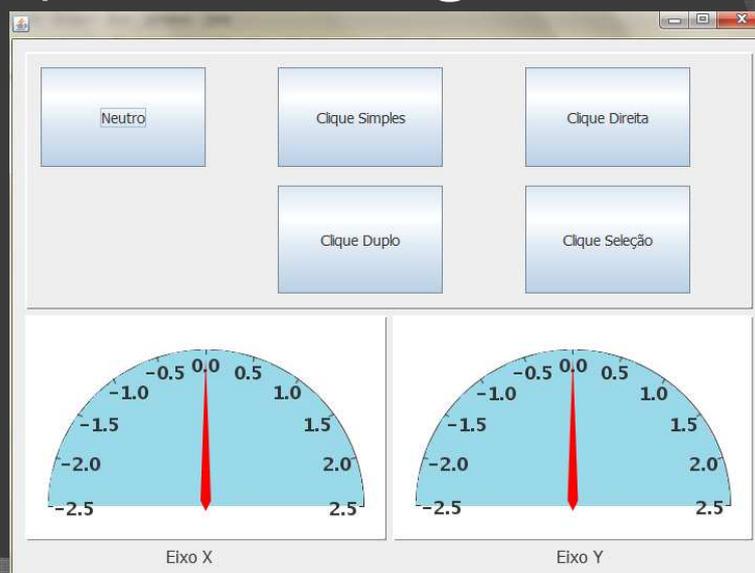
```

1. move(double valor, int positionOriginal) { (...)}
2.     if (valor >= neutroMin && valor <= neutroMax) {
3.         positionBefore = positionNow;
4.         positionNow = positionOriginal;
5.         positionClick = positionOriginal;
6.         clik = true;
7.         return;
8.     }
9.     double dif = 0;
10.    if (valor > neutroMax) {
11.        dif = valor - neutroMax;
12.        dif = Math.abs(dif);
13.        if (clik) {
14.            clik = false;
15.        }
16.    }
17.    (...)}
18.    positionNow = controlePosition(dif,
19.    positionOriginal);
20.}
```

Etapa 2: Manipulação do cursor – Movimentação do *mouse*

```
public void mouseMove() {  
    robo.mouseMove(x.positionNow,  
        y.positionNow);  
}  
  
public void mousePress(int botao) {  
    robo.mousePress(botao);  
}  
  
public void mouseRealise(int botao) {  
    robo.mouseRelease(botao);  
}
```

Etapa 3: Interface gráfica



Etapa 3: Interface

```
1. public calibrarEsquerdaSimples(Mouse mouse, String le) {
2.   if (mensagem) {
3.     JOptionPane.showMessageDialog(...);
4.     mensagem = false;     return;
5.   }
6.   mouse.calibrarEsquerdaSimples(le);
7.   if(mouse.click.calibrarEsquerda()){
8.     JOptionPane.showMessageDialog( ...);
9.   }else{
10.    JOptionPane.showMessageDialog( ...);
11.  } mensagem = true;
12.  return; }
```

Resultados

- Testes mostraram que os resultados são:
 - satisfatórios quanto a movimentação e calibração
 - insatisfatórios quanto a realização do clique
 - o mouse com acelerômetro pode ser indicado para pacientes com baixo e médio grau comprometimento

Conclusões

Extensão

- ◉ Substituir o SEN-00410 por outro kit que possua, além da leitura da aceleração gravitacional aplicada sobre o kit, a leitura das demais acelerações aplicadas sobre ele ou o resultado da somas das acelerações;
- ◉ Adicionar ao protótipo físico outro componente eletrônico que fique responsável por identificar cliques;
- ◉ Testar o protótipo em portadores de outras deficiências motoras;
- ◉ Montar um hardware que dispense a instalação de qualquer software no sistema operacional e que se conecte a USB;
- ◉ Permitir a calibração de velocidades de deslocamento independentes para cada uma das inclinações.



Demonstração



Obrigada