



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA Campus Vitória da Conquista

Sensor de Quedas

Sistemas Embarcados - 2015.2

Alexandre Sampaio e Italo Miranda

Apresentação

- Software que alerte para situações onde um objeto monitorado aparenta estar em queda livre.
- Em caso de queda, ter-se as altitudes inicial e final.
- Registrar *log* de informações dos sensores.

Sensores

GY-80: Multi sensor que dispões de 9 eixos:

- 3 para Giroscópio;
- 3 para Acelerômetro;
- 3 para Magnetômetro.

Ainda, 1 sensor de temperatura e pressão (barômetro).

Sensores

Para o projeto, foram utilizados:

- Sensor Digital Acelerômetro: ADXL345
 - Verificação de situação de queda.
- Sensor Digital de Pressão: BMP085
 - Verificação de altitude.

ADXL345

FREE_FALL

O bit FREE_FALL é setado (1) quando uma aceleração de menos que o valor definido no registrador THRESH_FF é experimentada por mais tempo que o definido no registrador TIME_FF.

* Os 3 eixos devem sofrer a aceleração correspondente à queda.

BMP085

Cálculo de Altitude Absoluta

Com a pressão medida a partir do próprio sensor (p) e a pressão ao nível do mar ($p_0 = 1013.25\text{hPa}$), a altitude em metros pode ser calculada a partir da fórmula barométrica internacional:

$$altitude = 44330 * \left[1 - \left(\frac{p}{p_0} \right)^{\frac{1}{5255}} \right]$$

Fonte: BMP085 Datasheet

Bibliotecas e Frameworks

- ADXL345 modificada por Bildr¹: biblioteca de licença pública geral (GPL), utilizada para acesso ao acelerômetro.
- Adafruit_BMP085²: biblioteca de licença BSD, escrita por Limor Fried e Ladyada para Adafruit Industries, utilizada para acesso ao barômetro.
- Log4j 2³: biblioteca Apache utilizada para a geração de logs da aplicação, altamente configurável.
- Java JNA⁴: framework usado para realizar a conexão entre a aplicação Java e a biblioteca dll.

¹Disponível em: <http://bildr.org/2011/03/adxl345-arduino/>

²Disponível em: <https://github.com/adafruit/Adafruit-BMP085-Library>

³Disponível em: <http://logging.apache.org/log4j/2.x/index.html>

⁴Disponível em: <https://java.net/projects/jna/downloads/directory/3.3.0>

Log4j 2

Diversos níveis de log disponíveis, inclusive a possibilidade de se criar os próprios níveis (o que não é muito recomendado).

TRACE < DEBUG < INFO < WARN < ERROR < FATAL

A ideia é representar mensagens de log da menor gravidade para a maior gravidade. Há métodos de logging para cada um desses níveis.

Desafios

- Entender as bibliotecas que continham métodos de detecção de queda.
- Entender a lógica para encontrar a altitude, a partir da pressão atmosférica.
- Cuidado com detalhes importantes na configuração do ambiente Eclipse para a geração da biblioteca dll/so.
- Entender e implementar a lógica de aplicação nas 3 camadas.
- Entender e utilizar a biblioteca Log4J.
- Particularidades do Linux.

Resultados

- ✓ Implementação e leitura de dados no Arduino realizada com sucesso
- ✓ Implementação da biblioteca sensorqueda.dll realizada com sucesso
- ✓ Implementação da App Java para a comunicação e tratamento dos dados oriundos das medições realizadas no Arduino realizada com sucesso

Conclusão

A complexidade e a quantidade de detalhes tornou o desenvolvimento trabalhoso. Ao mesmo tempo, o fato de ter que buscar alternativas, como as bibliotecas ADXL345 modificada por Bildr, e Adafruit_BMP085 proporcionaram o entendimento do funcionamento dos sensores.