XVIII SEPE:
SIMPÓSIO DE ENSINO, PESQUISA E EXTENSÃO
CIÊNCIA E INTERAÇÃO
PELA VIDA | 1 A 3 DE OUT/14

INTEGRAÇÃO JAVA COM ARDUINO

Alessandro A. M. De Oliveira[3], Alexandre O. Zamberlan[3], Reiner F Perozzo[3],
Rafael O. Gomes[1];Sergio R. H Righi[2],PecilcesP. Feltrin[2]

RESUMO

A integração de Linguagem de programação é importante para que se possa utilizar diversas linguagens em um software, este artigo tem o intuito de mostrar de forma técnica e detalhada a integração do Java com o Arduino, visando também a integração de sensores, onde se possa visualizar os dados no instantaneamente. Desta forma é possível utilizar vários equipamentos para a obtenção de dados que serão lidos, tratados e interpretados pelas linguagens em questão.

Palavras-chave: Automação; Arduino; Java;

1. INTRODUÇÃO

Muitos softwares dependem da coleta e tratamento de dados através de dispositivos externos, que por sua vez necessitam de uma linguagem diferente a qual o software utiliza usualmente, sendo assim é feita a integração das linguagens. Diversas linguagens de programação são utilizadas no desenvolvimento de software com isso existem vários desafios para integrar diferentes linguagens, para solucionar esses desafios existem técnicas para facilitar o desenvolvimento.

A linguagem Java é bastante utilizada no desenvolvimento de software, seu código é aberto e utilizado por um grande número de desenvolvedores. Também há muitas bibliotecas que facilitam o desenvolvimento, uma dessas bibliotecas é chamada de RXTX que será utilizada na integração Java com o Arduino.

Essa integração será realizada para que valores de um Encoder sejam recebidos pelo Arduino e enviados para uma aplicação que é desenvolvida em Java, para que esses dados sejam tratados.

[1] Acadêmico do Curso de Sistema de Informação – UNIFRA. rafael.degomes@gmail.com
[2] Acadêmico do Curso de Ciência da Computação – UNIFRA. feltrin10@gmail.com
[2] Acadêmico do Curso de Ciência da Computação – UNIFRA. a.r-1982@hotmail.com
[3] Professor do Curso de Ciência da Computação – UNIFRA. alessandroandre@unifra.br
[3] Professor do Curso de Ciência da Computação – UNIFRA. reiner.perozzo@gmail.com
[3] Professor do Curso de Ciência da Computação – UNIFRA. zamberlam@gmail.com

2. Ferramentas Utilizadas

Neste projeto foram utilizadas diversas ferramentas que serão abordadas a seguir.

2.1. Arduino

Arduino é uma placa de prototipagem eletrônica que integra *hardware* e *software* em sua placa ha um microcontrolador Atmega tem suporte para entrada e saída de dados, com uma linguagem de programação baseada em C, um dos objetivos da criação do projeto Arduino foi para que possa ser feito vários projetos com custos acessível e fácil de programar (Arduino, 2013). Existem várias extensões para o Arduino, como *Shields* de internet, GPS, GSM entre outros, A Figura 1 mostra o Arduino.

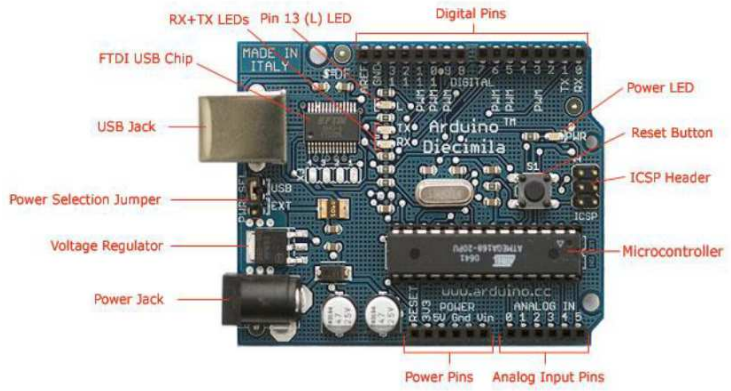

Figura 1: Arduino.

Fonte: ARDUINO. An open-sourceelectronicsprototypingplatform. Disponível em: <http://www.arduino.cc>. Acesso em: 11 fev.2014.

2.2. Java

Linguagem de Programação é orientada a objetos, foi desenvolvida em Chicago, a linguagem java não é compilada para códigos nativos mas sim compiladas para *bytecode* que essa compilação é feita por uma maquina virtual. Deve se levar em conta as suas características principais para criar códigos, é Orientada a Objetos, segurança, sintaxe é similar a C, varias bibliotecas prontas.

Como java é uma linguagem *OpenSource* ha muitas comunidades que ajudam no seu desenvolvimento, o java também tem uma grande vantagem a outras linguagem é a possibilidade de um mesmo *software* ser executa em diversas plataformas sob uma mesma compilação, a Figura 3 mostra um exemplo de código, esse exemplo cria uma classe animal e duas similares que quando são chamadas mostram o resultado do retorno da função.

```
public abstract class Animal {
   public abstract void fazerBarulho();
}

public class Cachorro extends Animal {
   public void fazerBarulho() {
      System.out.println("AuAu!");
   }
}

public class Gato extends Animal {
   public void fazerBarulho() {
      System.out.println("Miau!");
   }
}
```

Figura 3. Exemplo Código Linguagem Java

2.3. Biblioteca RXTX

Com a grande facilidade da linguagem Java de integrar diversas plataformas, são criadas bibliotecas para possibilitar o reaproveitamento do código, a biblioteca RXTX faz a comunicação tanto serial quanto paralela da porta USB, desta maneira os dados são enviados do Arduino para o aplicativo. Para iniciar a biblioteca basta importar para a IDE de desenvolvimento e iniciar a sua compilação. Seus códigos podem ser todos alterados fazendo assim com que tenha mais controle sobre a aplicação.

2.4. Encoder

Encoder é um sensor eletromecânico que conta e reproduz pulsos elétricos, conforme a forma que o eixo é rotacionado. Ele é utilizado para conversão de movimentos rotativos ou de deslocamentos que gera uma quantidade de pulsos por

volta. Com esses pulsos é possivel calcular as medidas de velocidade de rotação. Figura 6 mostra o Encoder.

Figura 6. Encoder
Fonte: Metalatex. **Guia Geral de Produtos Automação**.Disponível em < http://www.metaltex.com.br/guias/ga.pdf>.Acesso em 12 Jun. 2014.

3. INTEGRAÇÃO

Os dados gerados pelo Encoder são obtidos pelo Arduino, enviados atraves da biblioteca RXTX para a aplicação em Java.

```
15  /**
16   * Construtor da classe ControlePorta
17   * @param portaCOM - Porta COM que será utilizada para enviar os dados para o arduino
18   * @param taxa - Taxa de transferência da porta serial geralmente é 9600
19   */
20  public ControlePorta(String portaCOM, int taxa) {
21    this.portaCOM = portaCOM;
22    this.taxa = taxa;
23    this.initialize();
24  }
25
26  /**
27   * Médoto que verifica se a comunicação com a porta serial está ok
28   */
29  private void initialize() {
30    try {
31      //Define uma variável portId do tipo CommPortIdentifier para realizar a comunicação
32      CommPortIdentifier portId = null;
33      try {
34        //Tenta verificar se a porta COM informada existe
35        portId = CommPortIdentifier.getPortIdentifier(this.portaCOM);
36      }catch (NoSuchPortException npe) {
37        //Caso a porta COM não exista será exibido um erro
38        JOptionPane.showMessageDialog(null, "Porta COM não encontrada.",
39                  "Porta COM", JOptionPane.PLAIN_MESSAGE);
40      }
41      //Abre a porta COM
42      SerialPort port = (SerialPort) portId.open("Comunicação serial", this.taxa);
43      serialOut = port.getOutputStream();
44      port.setSerialPortParams(this.taxa, //taxa de transferência da porta serial
45                               SerialPort.DATABITS_8, //taxa de 10 bits 8 (envio)
46                               SerialPort.STOPBITS_1, //taxa de 10 bits 1 (recebimento)
47                               SerialPort.PARITY_NONE); //receber e enviar dados
48    }catch (Exception e) {
49      e.printStackTrace();
50    }
51  }
```

Figura 7. Código da Biblioteca RXTX

Após a instalação das *IDEs* tanto do Arduino quanto a do Java foi também importada a biblioteca RXTX. A Figura 7 mostra parte do código da biblioteca, esta mostra a comunicação com a porta serial, que é responsável por enviar os dados do arduino para o Java. Uma ves definida a taxa de transferência em bauds, e a porta serial o *software* ira iniciar a comunicação entre os dois componentes, e mostrara na tela os dados que estão sendo obtidos pela porta serial.

No arduino foi feita a inicialização das variáveis do encoder, para que possa ser recebido pela porta serial e enviado para a aplicação. Foi utilizado o Serial Print para que se possa ser visualizado, na Figura 8 é mostrado parte do código do encoder.

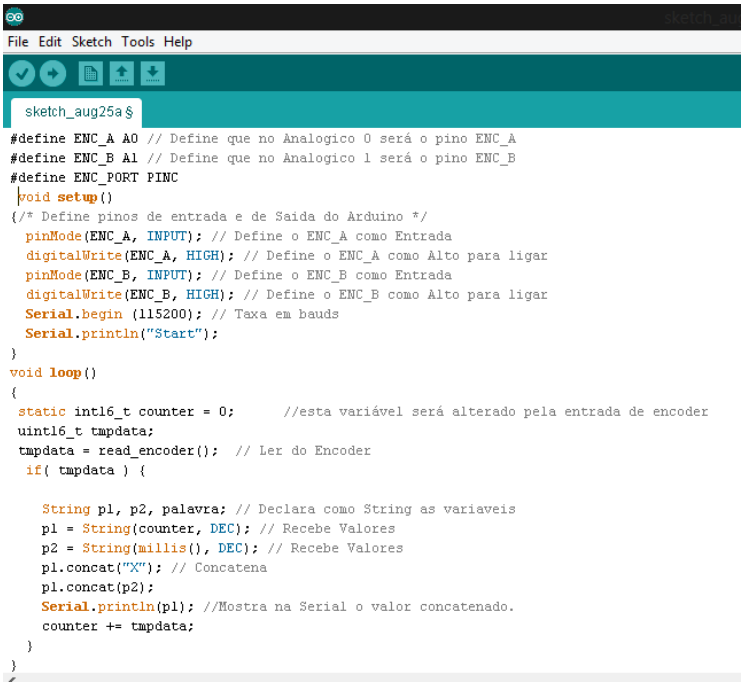

Figura 8: Codigo para receber os dados do Encoder.

4. CONCLUSÃO

É possível fazer a integração de muitas linguagens de programação e de vários dispositivos para que se tenha um melhor desempenho.

Existem várias formas de chegar até essas integração, neste trabalho utilizou a integração do Arduino com o Java, possibilitando assim disponibilizar os dados adquiridos pela plataforma Arduino em outra linguagem de programação.

REFERÊNCIAS

ARDUINO. An open-sourceelectronicsprototypingplatform. Disponível em: <http://www.arduino.cc>. Acesso em: 17 out.2012.

THOMAZINI, Daniel. ALBUQUERQUE, Pedro U. B. Sensores industriais – Fundamentos e Aplicações. 5º ed. São Paulo: Érica,2005 . 222p.

SABER ELETRONICA, São Paulo: Editora Saber, n. 405, out. 2006.

DEVMEDIA. Utilizando a API RXTX para manipulação da serial – Parte III Disponível em: < http://www.devmedia.com.br/utilizando-a-api-rxtx-para-manipulacao-da-serial-parte-iii/7171>. Acesso em: 15 julho.2014.

G. Cornell, C. S. Horstmann, Métodos Nativos, São Paulo, 2003, pp. 755-785.

HARVEY M. Deitel. Java: Como Programar. 6 ed. São Paulo: Pearson education do Brasil, 2005.

ROSÁRIO, João Maurício, Princípios de Mecatrônica, São Paulo, Prentice Hall, 2005.