

FIA/P GRADUAÇÃO

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Arquiteturas Disruptivas, IoT, Big Data e IA

Prof. Rafael Matsuyama

Roteiro

- ✓ Protocolo Firmata
 - ✓ Exercícios de JSON
 - ✓ Arquitetura Cliente / Servidor
 - ✓ Protocolo Firmata
 - ✓ Firmata4J

Internet das Coisas

Exercícios de JSON (<http://www.jsonlint.com>) (Pt. 1)

1. Usando o Notepad++, crie um objeto JSON contendo os seguintes campos:
 - nome (string)
 - sobrenome (string)
 - cpf (string)
 - nascimento: objeto contendo os campos numéricos:
 - dia
 - mes
 - ano
 - endereco: objeto com os seguintes subcampos
 - logradouro (string)
 - numero (numérico)
 - cep (string)
 - cidade (string)
 - estado (string)

Exercícios de JSON (<http://www.jsonlint.com>) (Pt. 2)

2. Usando outro arquivo, crie um documento contendo os seguintes campos
 - cpf (string): cpf do cliente que fez a compra
 - hora (string data-hora): quando foi feita a compra
 - itens (array): lista de identificadores numéricos dos produtos comprados
 - preco (numérico): preço pago pelas mercadorias

Arquitetura Cliente / Servidor (Pt. 1)

- Embora a programação do Arduino seja simples, usando uma interface intuitiva, podemos querer trocar a programação da placa após sua implantação no seu local de aplicação.
 - Por exemplo, podemos ter instalado uma Arduino Nano em um local de difícil acesso para controlar uma lâmpada ou medir a quantidade de água consumida pelo chuveiro.
 - Mudar a programação pode significar mudar a porta de leitura do fluxo de água ou ainda acrescentar uma nova funcionalidade, como controlar a temperatura por PWM.

Arquitetura Cliente / Servidor (Pt. 2)

- Para evitar essa situação, podemos deixar que a programação do Arduino apenas processe comandos vindos de fora, funcionando como um pequeno servidor genérico que responde a requisição de clientes.
- Assim, as ações específicas da placa Arduino são determinadas pelo cliente externo, e eventuais mudanças devem ser programadas nesse cliente.

Protocolo Firmata (Pt. 1)

- O protocolo Firmata permite que um micro controlador receba mensagens padronizadas para que ele execute certas tarefas, tais como:
 - Leitura de porta analógica (ADC).
 - Acionamento de servo motor em um ângulo específico.
 - Acionamento de LED, etc.
- Usando o protocolo Firmata, uma placa Arduino pode funcionar um servidor respondendo a comandos de uma cliente externos através de comunicação serial, seja através de uma porta serial ou outros dispositivos, como Wi-Fi, Bluetooth, etc.

Protocolo Firmata (Pt. 2)

- Desvantagem: certas funcionalidades que dependem de bibliotecas específicas, como a leitura do sensor DHT, podem não estar disponíveis nativamente no protocolo Firmata!
- Para o Arduino rodar o protocolo Firmata, basta carregar o programa:

File >> Examples >> Firmata >> StandardFirmata

Como Conectar com o Firmata? (Pt. 1)

- Se o programa servidor usando o protocolo Firmata roda no Arduino, que cliente podemos usar?
- Para o Node.js, podemos usar o pacote Johnny Five.
 - Nome dado em homenagem ao robô do filme “Short Circuit: O Incrível Robô”.
 - Framework para programação de dispositivos que aceitam o protocolo Firmata.
 - Para os demais dispositivos, adaptadores chamados “IO-plugins” precisam ser usados.



Como Conectar com o Firmata? (Pt. 2)

- Muito usado para programar kits de robótica com base em placas Arduino.
- O Johnny Five é uma API para Node.js. Para Java temos o Firmata4J.



Exemplo do Firmata4J (Pt. 1)

```
public static void main(String[] args){
    IODevice device = new FirmataDevice("COM3");
    // Faz a subscrição a eventos com
    device.addEventListener(...);
    // e/ou device.getPin(n).addEventListener(...);
    device.addEventListener(new IODeviceEventListener()
    {
        @Override public void onStart(IOEvent event) {
/*...*/ }

        @Override public void onStop(IOEvent event) {
/*...*/ }
    });
}
```

Exemplo do Firmata4J (Pt. 2)

```
@Override public void onPinChange(IOEvent
event) {
    // Aqui é indicada a mudança de valor da
    porta

    Pin pin = event.getPin();
    System.out.println( String.format(
        "Porta %d tem valor %d",
        pin.getIndex(),
        pin.getValue() )
    );
}
```

Exemplo do Firmata4J (Pt. 3)

```
@Override public void onMessageReceive(IOEvent
event, String message) {
    //Aqui são processadas as mensagens vindas
do dispositivo
    System.out.println(message);
}
}
device.start(); // inicializa a comunicação com o
device
device.ensureInitializationIsDone(); // espera
pela total inicialização
// loop de processamento do programa
while(...) { /* ... */ }
device.stop(); // Finaliza a comunicação com o
dispositivo
```

Trabalhando com o Firmata em Java (Pt. 1)

- Primeiro, é necessário instalar o servidor Firmata no Arduino.
 - Existem várias versões, mas para o básico precisamos apenas do StandardFirmata, já fornecida pela Arduino IDE.
- Então, é necessário promover uma comunicação de caracteres entre o Arduino e o host com o cliente Firmata, seja com ou sem fio.
 - Vamos usar a comunicação serial via cabo USB; para outras formas de comunicação algumas instruções devem ser seguidas.

Trabalhando com o Firmata em Java (Pt. 2)

- Firmata4J tem as funções básicas para as atividades fundamentais das placas Arduino:
 - Leitura Analógica (ADC).
 - Leitura e escrita de porta digital.
 - Controle PWM de dispositivos (saída “analógica”).
- Instalando como dependência do Maven:
 - **groupId**: com.github.kurbatov
 - **artifactId**: firmata4j
 - **version**: 2.3.6

REFERÊNCIAS



1. http://www.telecom.uff.br/pet/petws/downloads/tutoriais/arduino/Tut_Arduino.pdf
2. <http://arduino.cc/en/Reference/HomePage>
3. http://www.firmata.org/wiki/Main_Page
4. <https://github.com/kurbatov/firmata4j>

Copyright © 2018-2019 Prof. Rafael Matsuyama / Prof. Antônio Selvatici

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).