Sumário

[Método void Setup() 3](#_Toc464107365)

[Entrada e Saída 3](#_Toc464107366)

[void PinMode(int pino, String modo) 3](#_Toc464107367)

[Void Serial.begin(int velocidade) 3](#_Toc464107368)

[Método void Loop() 3](#_Toc464107369)

[void digitalWrite(int pino, bool valor) 3](#_Toc464107370)

[int digitalRead(int pino) 3](#_Toc464107371)

[int analogRead(int pino) 3](#_Toc464107372)

[int map(int valor, int deMinimo, deMaximo, paraMinimo, paraMaximo) 3](#_Toc464107373)

[void Serial.print(String valor) 3](#_Toc464107374)

[Interface Java X Arduino 4](#_Toc464107375)

[Arduino 7](#_Toc464107376)

# Método void Setup()

Este método permite a configuração e inicialização do I/O do Arduino. Ele é executado no primeiro scan.

## Entrada e Saída

### void PinMode(int pino, String modo)

Configuração do pino especificado para se comportar como entrada ou saída. O parâmetro pino deve ser do tipo inteiro e o parâmetro modo deve ser uma das duas constantes: INPUT ou OUTPUT.

### Void Serial.begin(int velocidade)

Configuração da comunicação serial. O parâmetro velocidade é a taxa de transferência de dados entre o recebtor e o transmissor.

# Método void Loop()

Este método realiza loops consecutivos, permitindo o programa atualizar e responder. Todo o controle do Arduino é realizado dentro deste método.

### void digitalWrite(int pino, bool valor)

Seta um pino configurado como OUTPUT. O parâmetro pino referencia a um pino configurado em setup() como OUTPUT. O parâmetro valor é uma das duas constantes: HIGH ou LOW. O padrão é HIGH.

### int digitalRead(int pino)

Lê um valor de um pino específico. O retorno é uma String com uma das duas constantes: HIGH ou LOW. O parâmetro pino é o pino referenciado que deseja-se o valor.

### int analogRead(int pino)

Lê o valor analógico de um pino definido pelo parâmetro pino. O valor lido é um inteiro de 10 bits (0 a 1023 bits). A frequência máxima de leitura de transição é de 10kHz. O valor de resolução é .

### int map(int valor, int deMinimo, deMaximo, paraMinimo, paraMaximo)

Passa o valor numérico de uma base linear(escala) para outra escala.

### void Serial.print(String valor)

Escreve um valor na porta serial. Este valor pode ser monitorado pelo Serail Monitor da IDE do Arduino.

# Interface Java X Arduino

1. Usando a IDE do Arduino se parametriza a comunicação serial na porta correta.



1. Instalar a dll e a jar necessárias para usar o hardware do PC.

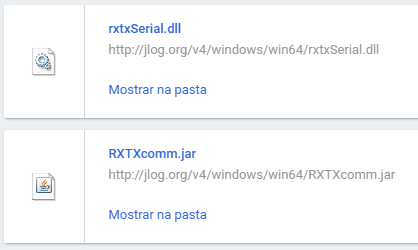


Figura - Arquivos necessários para comunicação serial Java ao Arduino

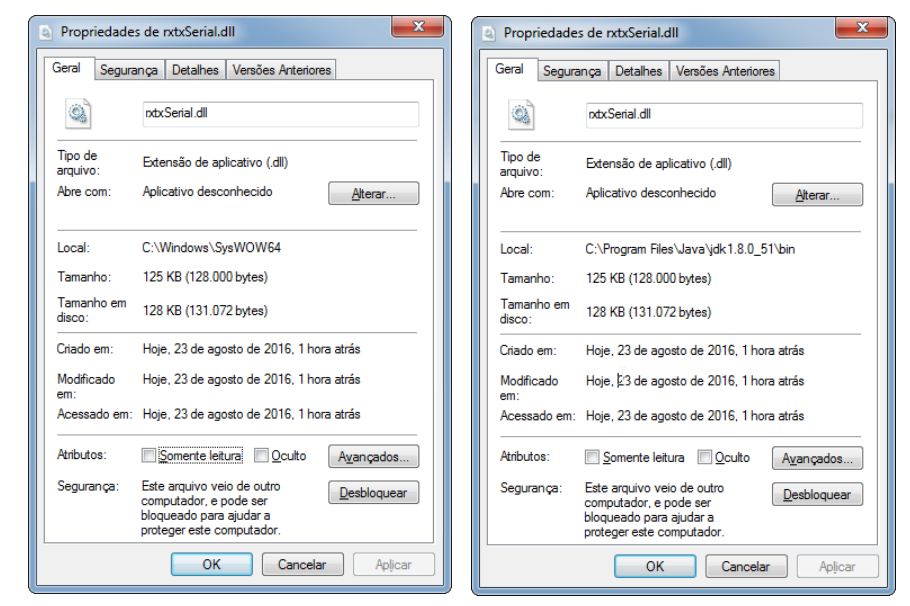


Figura - Arquivos para serem colocados no classpath

1. Copiar a jar necessária para implementar a interface.

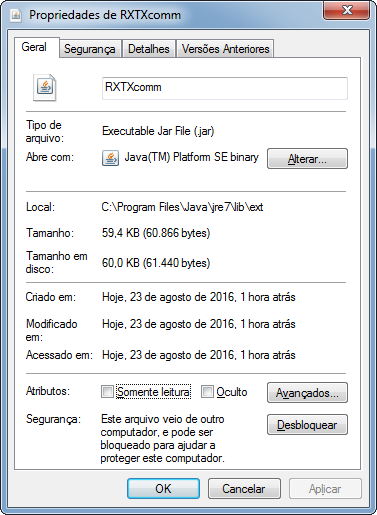


Figura - JAR necessária para comunicação serial com o Arduino

1. Na classe de acesso à comunicação serial basta escrever o código para acessar os pinos do Arduino. Os pinos devem estar já declarados usando a IDE do Arduino ou outra qualquer.

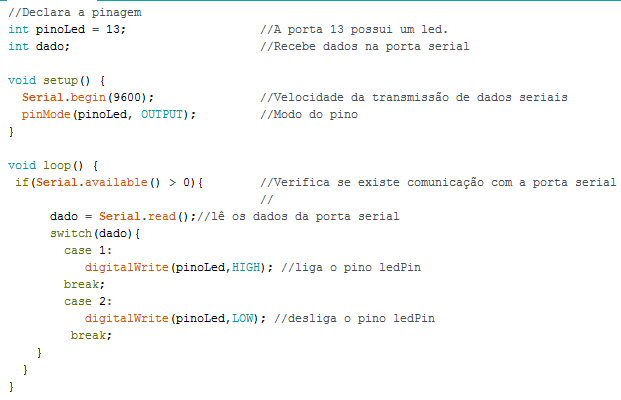


Figura 4 - Sketch na IDE do Arduino

FUTURO

Fazer a pré-configuração através do Simulink

Fazer a escrita e então a leitura de valor analógico

Verificar o tempo de comunicação

Montar shield Ethernet e tentar comunicar

Fazer o JSP para poder ler do e escrever no Arduino

Fazer o CSS para mobile

## Arduino