Communicate with Arduino through Java

Arduino can send message to the serial port, use Java to get the serial messages.

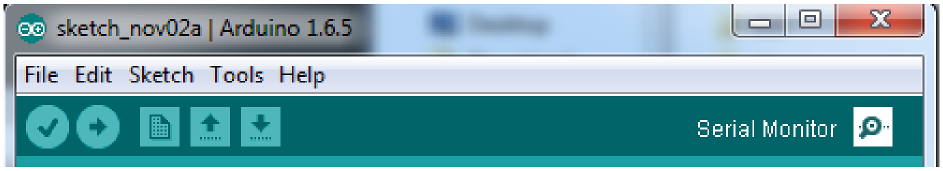
1. Code on Arduino:

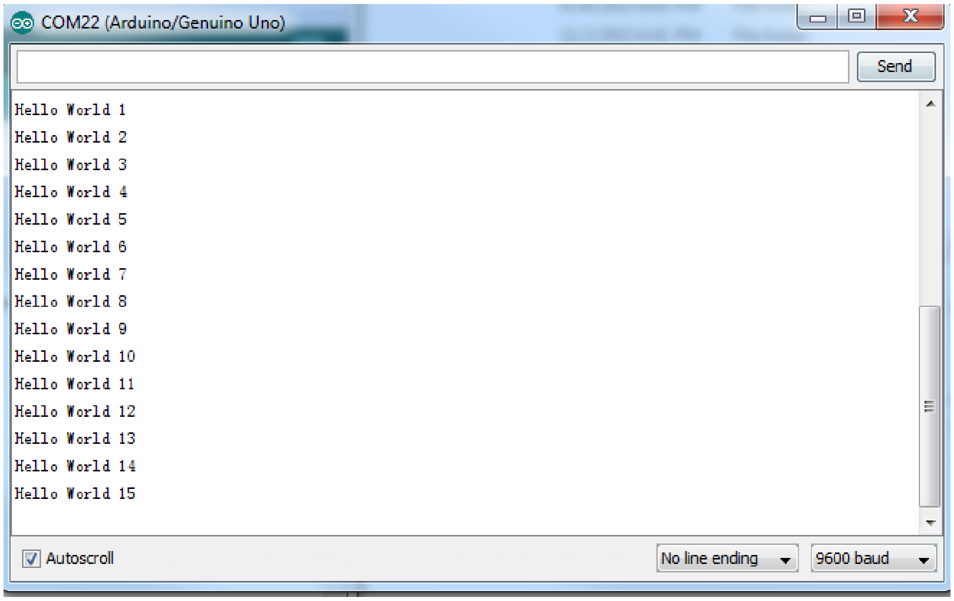
```
void setup(){
    Serial.begin(9600);
}
void loop(){
    Serial.println("Hello world");
    delay(1000);
}
```

Upload this piece of code to Arduino, Arduino would send the string “Hello World” to the serial port for every 1 second.

2. Use Serial Monitor of the Arduino IDE to see the serial content

You can use the Serial Monitor of the Arduino IDE to check the serial content that are sent by Arduino.





```
Hello World 1
Hello World 2
Hello World 3
Hello World 4
Hello World 5
Hello World 6
Hello World 7
Hello World 8
Hello World 9
Hello World 10
Hello World 11
Hello World 12
Hello World 13
Hello World 14
Hello World 15
```

☑ Autoscroll

No line ending

9600 baud

3. Download Java serial communication library

For 32-bit machines (Windows, Mac OX, and Linux), download from here:

http://rxtx.qbang.org/pub/rxtx/rxtx-2.1-7-bins-r2.zip

For 64-bit machines (Windows and Linux), download from here:

http://fizzed.com/oss/rxtx-for-java

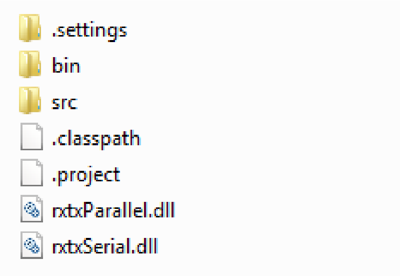
For 64-bit Mac:

http://blog.iharder.net/2009/08/18/rxtx-java-6-and-librxtxserial-jnilib-on-intel-mac-os-x/

4. Include the serial library to your project

Unzip the downloaded file, put the *RXTXcomm.jar* to *$JAVA_HOME$/lib/ext*.

For windows, put the *rxtxSerial.dll* and *rxtxParallel.dll* to the root directory of your java project Eclipse.



.settings
bin
src
.classpath
.project
rxtxParallel.dll
rxtxSerial.dll

For other machines, referred to the following link for installation instructions:

http://playground.arduino.cc/Interfacing/Java

5. Run the example code

In Eclipse, create a java project, and use the sample Java code from the following link:

http://playground.arduino.cc/Interfacing/Java

You may need to change the serial port names depending on your machine. Modify this piece of code to match the serial port name of the Arduino on your machine:

```
private static final String PORT_NAMES[] = {
        "/dev/tty.usbserial-A9007UX1", // Mac OS X
        "/dev/ttyUSB0", // Linux
        "COM4", // Windows
};
```

Run the code, you will see the string “Hello World” printed on the console.

The piece of code that actually reads the serial port content is the following function:

```
    public synchronized void serialEvent(SerialPortEvent oEvent) {
        if (oEvent.getEventType() == SerialPortEvent.DATA_AVAILABLE) {
            try {
                String inputLine=input.readLine();
                System.out.println(inputLine);
            } catch (Exception e) {
                System.err.println(e.toString());
            }
        }

    }
```

When Arduino sends sensor readings to the serial port, you will get the sensor readings from this function.

Reference:

Arduino and Java: http://playground.arduino.cc/Interfacing/Java