MIT AITI Mobile Development in Java
SMS Hardware Set-Up Information

A USB GSM Modem is an example of a direct to mobile gateway. The modem is a device that allows you to send and receive SMS messages; it directly connects to the operator's network. It interfaces with your computer through the USB port. In order for the modem to communicate with your Java SMS application, you must install the modem's driver and copy the libraries that Java uses to communicate with the modem. Follow the following steps to set up the GSM modem and appropriate files.

Step 1: (Installing Driver)

Install the correct driver for the modem you will be using. This will differ depending on the modem you are using. Either:

- Insert the CD bundled with the modem, follow the directions from the autorun application, or
- Some modems create a fake CD drive with the software, double click to this drive to autorun the install application.

After driver installation is complete, insert the GSM modem into one of the USB ports and make sure that it is recognized correctly.

Step 2: (Placing files in correct directory)

Download the needed SMSLib files from the e-learning site. They are placed in three different folders (lib, lib_ext, and bin). The files in these folders will be copied into their respective folders in the latest version of the JRE you are using. To access the folder of the JRE, go to the C drive, then "Program Files", then open up the "Java" folder. There might be multiple folders in this folder, but select the latest version of the JRE.

When you open the JRE, you will see two folders, bin and lib, and maybe 10 other files. You will then place all the files from the lib folder that you downloaded into the lib folder in the JRE folder. You will also do the same for the files in the bin folder. As for the files in the lib_ext folder, you copy these files to the lib\ext directory of your JRE.

Additional information about the files you are installing:

Java Comm is a java extension that facilitates communication technologies (in our case the modems) and allows access through USB ports. To read more about the API, visit http://java.sun.com/products/javacomm/. The following files were part of this installation:

- File comm.jar should go under JDKDIR/jre/lib/ext/
- File javax.comm.properties should go under JDKDIR/jre/lib/
- Library files (i.e. win32com.dll for Win32 or the .so Linux library files) should go under JDKDIR/jre/bin/

RxTx is a native library for serial and parallel communication in Java. It provides a portable implementation of Sun's JavaCOMM. The following files were part of this installation:

- File RXTXcomm.jar should go under JDKDIR/jre/lib/ext/
- The necessary library (e.g.. for Linux 32bit, the librxtxSerial.so) should go under JDKDIR/jre/bin/

Step 3: (JAR FILE)

Download the org.aiti.sms.Main.java and aiti-sms.jar file from the e-learning website. Create a new package in your Java project (File→New→Package) and name it org.aiti.sms. Then import the Main.java file into this package by going to file→import→general→file system and then browse and select the file from wherever you saved it. Next, import the aiti-sms.jar file by right-clicking on the Java project and selecting Properties. Then, on the left hand side, select "Java Build Path". Then select the "Libraries" tab on the top, then the "Add External Jars…" button on the right hand side. Browse and select the aiti-sms.jar file and click OK. The Main.java file and the aiti-sms.jar file should now appear in your Package Explorer.

Step 4: (Setting the Handler)

Depending on the name of the file of your application, you have to change the name of the new object created in the handler. In the Main.java file, this occurs in line 36, which reads:

```
SMSHandlerThread.setAITIInboundMessageNotification(new SimpleApp());
```

You must change the object created (in this case SimpleApp) to the name of your class that services inbound messages. Each time an inbound SMS message is received, a new thread is spawned and a new object of you handler type is created. This new object then handles the inbound message in its own thread. Remember, your handler class must implement the `IAITIInboundMessageNotification`.

Step 5: (COM PORT)

If you are using a GSM USB modem, you will need to set the COM port that the modem is using. The easiest way to detect the COM port number is to run the command "mode" in the

command line before you insert the modem. Then insert the modem and run the command "mode" again. Simply observe which COM port was added and this additional COM port is the COM port number through which the modem is connecting. Line 42 in the Main.java file reads

```
app.setComPort((short)7, 460800);
```

The number following `(short)` is the COM port number and should match the COM port of the modem. The second argument is the port speed; you usually do not have to change this setting.

Step 6: (OPTIONAL: SETTING PROXY SERVER SETTINGS)

If you must connect to the Internet via a proxy server, you must uncomment line 39 in the Main.java file. It will read:

```
//app.setProxyServer("192.168.170.25", 3135);
```

Change the HTTP proxy address and port if necessary (they can be found in connection settings in the Internet options of your browser).

Step 7: (OPTIONAL: USING EMULATOR)

If you would like to use the emulator, you must change line 33, which reads:

```
AITISMSServer app = new AITISMSServer();
```

Pass the boolean constant `true` to the constructor to enable the emulator:

```
AITISMSServer app = new AITISMSServer(true);
```