

The article below is dedicated to the description of the possibility to use COM port in applications for OS Android. It is needed when your application must receive/send data from the other device having only Serial port RS 232 as means of communication.

What do we need?

First, it is necessary to download Android NDK to work with native code in Java.

<http://developer.android.com/tools/sdk/ndk/index.html>

It will be necessary to call code on C to work with the serial port; Second, it is necessary to check-out source code of the project from svn to work with Serial port RS 232.

svn

checkout

<https://code.google.com/archive/p/android-serialport-api/>

Organization of the project

It is necessary for us to create a folder **/jni** and to copy the contents of a folder **/jni** of the project, downloaded from svn (or simply to copy all folder **/jni**), in Android project. After that we need to add the following files from the downloaded project:

- SerialPort.java
- Application.java
- SerialPortActivity.java
- SerialPortFinder.java

The given files allow us to work with the COM port but you will need to edit them to serve the needs of the project. It is known all devices in the systems similar to a *nix one are on a path **/dev**. To establish **path to the device (COM port)** and **baud rate** it is necessary to set appropriate values of fields in Application.java file:

```
String      path      =      -path      to      +      device-;
int baudrate = -baud rate-;
```

Class **SerialPortActivity.java** is an extension of **Activity** class and contains an abstract method **protected abstract void onDataReceived (final byte [] buffer, final int size)**. You can inherit the Activity from this class where there will be work with Serial Port, and to process data acquisition from the port having redefined the method **onDataReceived**.

```
@Override
protected void onDataReceived(final byte[] buffer, final int size) {
    runOnUiThread(new Runnable() {
        public void run() {
            //TO DO your log
```

```
}
}
}
}
```

It is possible to extract logic of information system of port in your own class/classes and not to use the inheritance from **SerialPortActivity.java**. You can receive the lists of all devices and their paths by means of class **SerialPortFinder.java** with its methods **getAllDevices ()** and **getAllDevicesPath ()** accordingly. Write-In port is carried out by using a simple record in **OutputStream** created by means of class **SerialPort.java**, COM port.

```
mOutputStream.write(new
mOutputStream.write('\n');
```

```
String("text").getBytes());
```

JNI & NDK

Native code loading in Android application may be found in class **SerialPort.java** by calling **System.loadLibrary (' serial_port ')**:

```
private native static FileDescriptor open(String path, int baudrate, int flags);
public native void close();
static {
    System.loadLibrary("serial_port");
}
```

The parameter **serial_port** is a module that has occurred as a result of code C compilation through/via NDK. It is specified in file make **/jni/Android.mk**. File **SerialPort.c** in **/jni** folder contains native functions of the system calls to work with Serial port. Such parameters for COM port as Data **bits**, **Parity**, **Stop bits** and the others may be changed in this file by means of structure **termios**, for example:

```
cfg.c_cflag      &=      ~CSTOPB;
cfg.c_cflag      &=      ~CSIZE;
cfg.c_cflag      |=      CS8;
(Data bits=8, Parity=none, Stop bits=1)
```

This information about **termios.h** may be found under the following link: <http://pubs.opengroup.org/onlinepubs/007908775/xsh/termios.h.html> After changing the file **SerialPort.c** it is necessary to compile libraries as follows:

- Open command line
- Go to the NDK folder
- Set path to the Android project – **set NDK_PROJECT_PATH= -path to your android project-**
- Run – **ndk-build**

The libraries will be compiled and added in a folder **/libs** of your project. After that it is possible to perform the following command in order to install Android application in a device:

adb install -path to you .apk file-