Building/Installing the Test Programs

The test programs are comprised of Python based PC programs that communicate with a Java based Android Application that runs on an Android table or device. The client programs make use of the Python Serial library to communicate through either Linux (or Unix derivatives) based "tty" devices or Windows based COM ports.

Unpacking the Files

You should unpack both tar.gz files into the same root directory. Here's an example in Linux.

```
shell > mkdir viasat_programs
shell > cd viasat_programs
shell > tar -zxf AIOUARTLoopback.tar.gz
shell > tar -zxf SerialClientTest.tar.gz
shell > ls
AIOUARTLoopback/  AIOUARTLoopback.tar.gz  SerialClientTest/  SerialClientTest.tar.gz
```

Build Application with Apache Ant

There are two ways to go about building the application. The easiest way is if you have already added your android SDK to your path and if you have Ant installed. First you would

```
shell > cd AIOUARTLoopback
shell > ant debug
shell > adb install bin/com.UARTLoopback.UARTLoopbackActivity-debug.apk
```

Build Application with Eclipse

In this case you will need to create a new project (New -> Other -> Android Project from Existing Code). Specify the directory of AIOUARTLoopack for the import. After importing , perform a "clean". Create a "Run Configuration" referencing an API level of 13. followed by a "build".

Running the Smoke Test

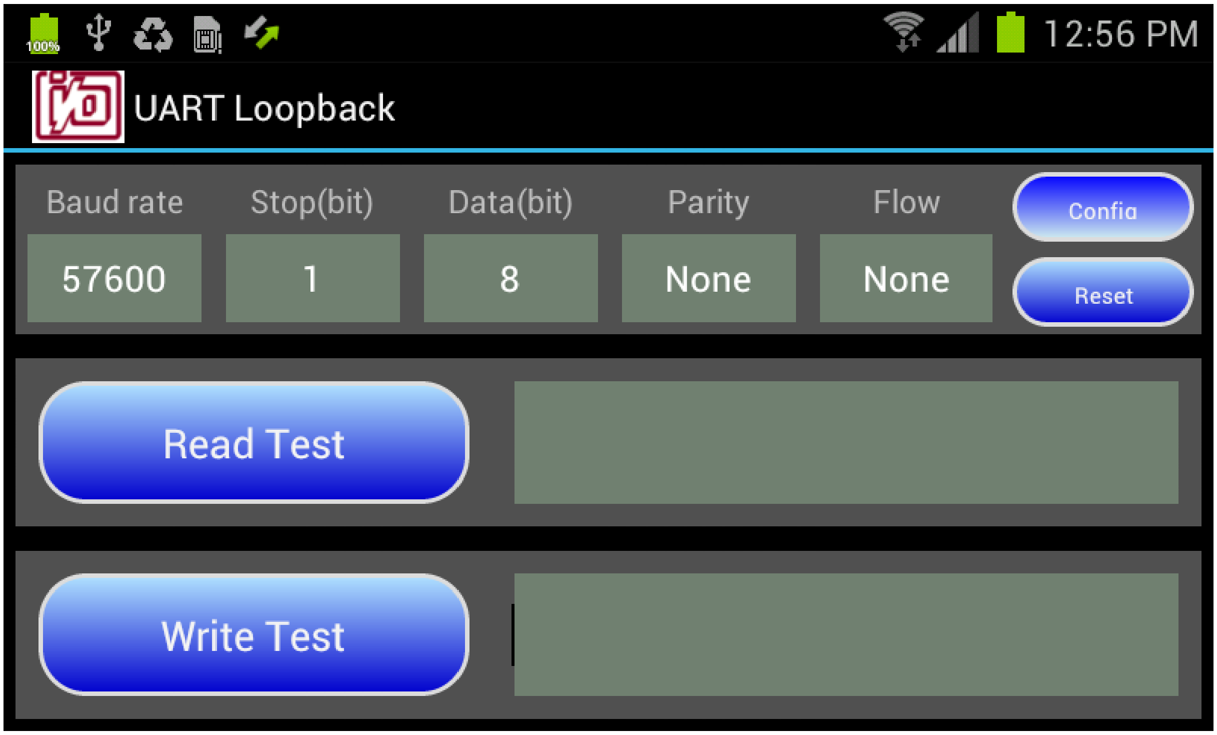
You will want to follow the procedure of starting the read test followed by the corresponding write test on the opposite platform with the reason being that you will want to perform a comparison between the bytes written from one end to the number of bytes received on the opposite end. This can only be achieved reliably if you stop the writer process first.

Start the Android Application

If you plug in the FTDI chip PCB, the android application will recognize it and prompt you with a message saying "**Open UART Loopback when this USB accessory is connected**" . Click the button that says **Use by default for this USB accessory**.

Configure the modem

In this set of examples provided in this document we set it to 57600 and the press the "Config button"

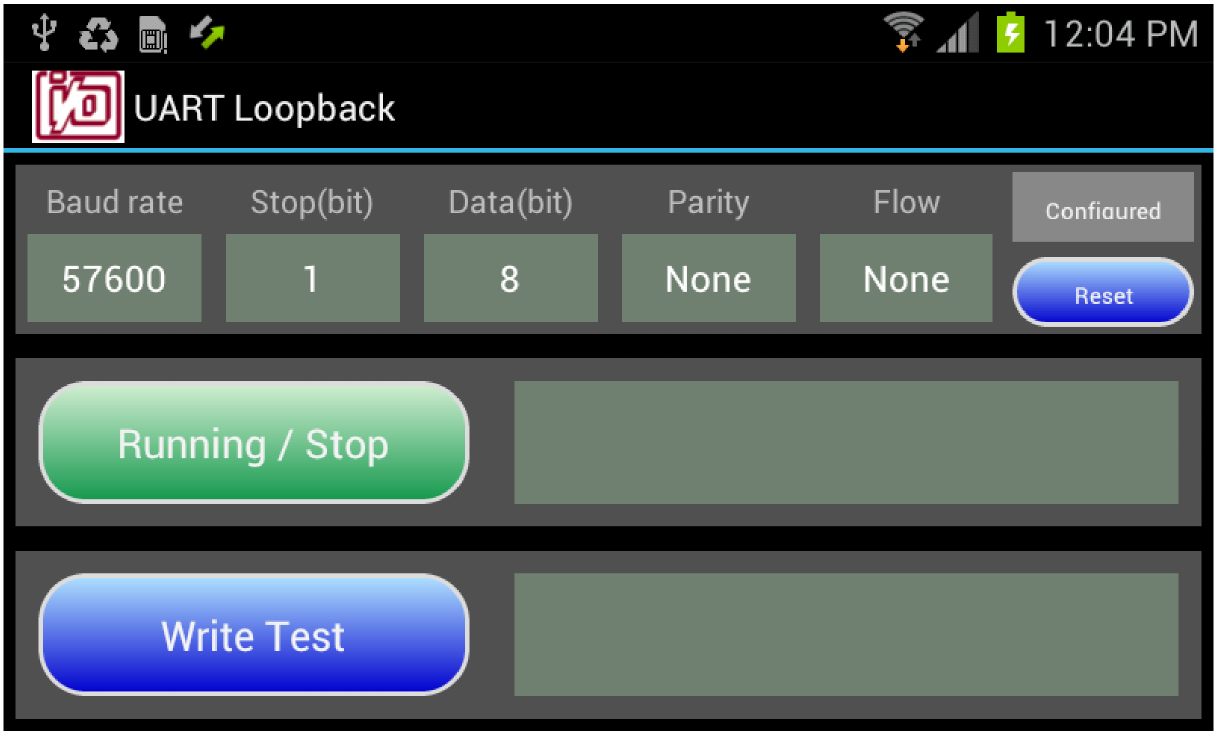


Baud rate	Stop(bit)	Data(bit)	Parity	Flow
57600	1	8	None	None

Figure 1: Configuring the Modem parameters

Start the Read Test

You will now start the Read Test on the android device. You will accomplish this by pressing the button marked "Read Test" You should see the following image

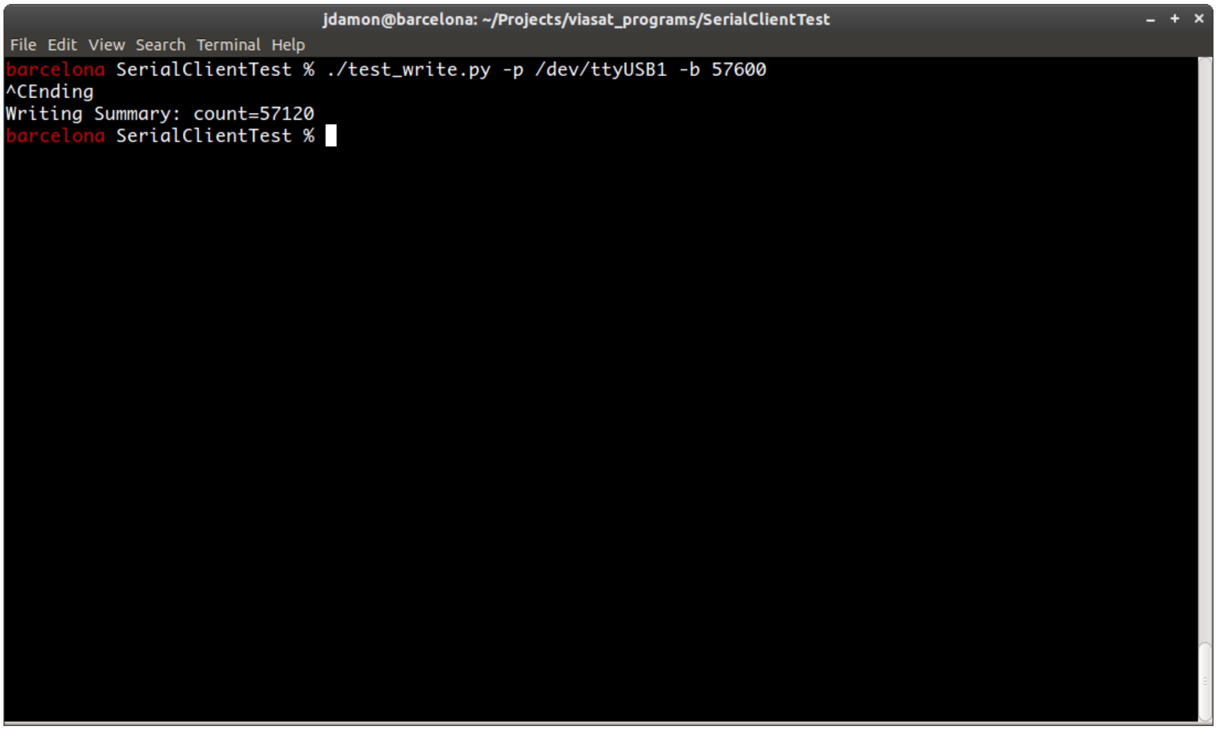


Baud rate	Stop(bit)	Data(bit)	Parity	Flow
57600	1	8	None	None

Figure 2: Read Test in progress

Perform Write Test

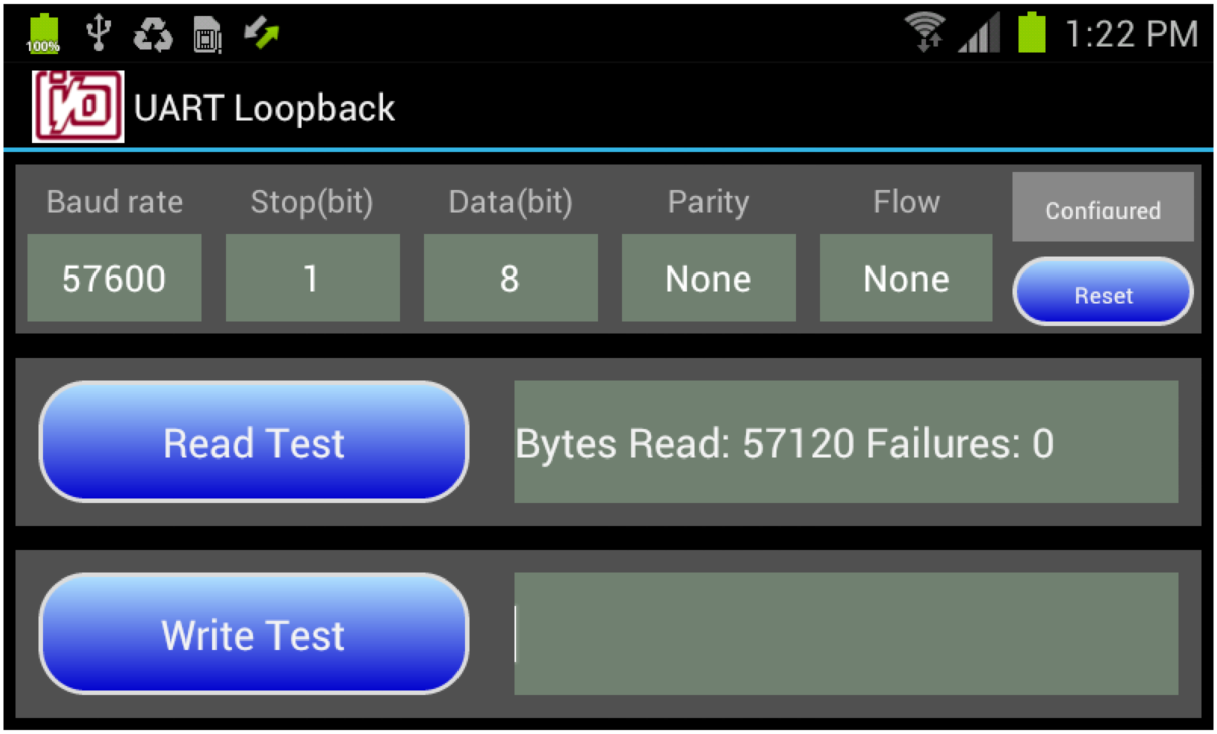
You will Start the write test from the PC and hit Control-C to interrupt it after it has run for a time period that you see fit. At first you might want to run it after a second or two. While the Write Test is running you will see bytes being written to the text display right next to the Read Test button.


jdamon@barcelona: ~/Projects/viasat_programs/SerialClientTest
File Edit View Search Terminal Help

```
barcelona SerialClientTest % ./test_write.py -p /dev/ttyUSB1 -b 57600
^CEnding
Writing Summary: count=57120
barcelona SerialClientTest %
```

Figure 3: Client finishing writing packets

5. Stop the Reading test on the android device and you should see a picture as follows with a line saying “Bytes Read: “ and a number matching the count displayed in the PC client’s window.



UART Loopback

Baud rate	Stop(bit)	Data(bit)	Parity	Flow
57600	1	8	None	None

Bytes Read: 57120 Failures: 0

Figure 4: Finished Read test

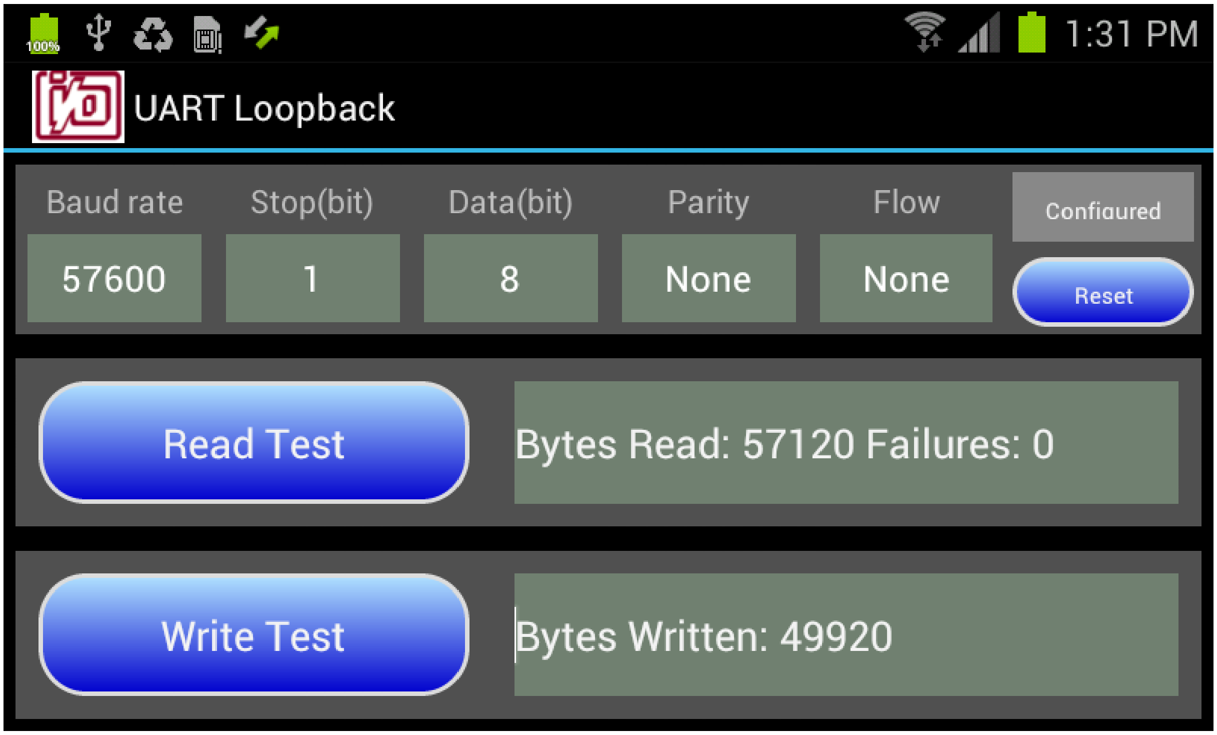
Continuing with testing

At this point you should next follow up running the Write Test program where the Android device performs an endless loop of Test packets and the PC client program “test_read.py” will read from the serial connection and display both the count as well as any read errors that were encountered.

To perform the Read test from the client , just start the write test on the Android device by pressing “Write Test”. Then on the PC you run the program as follows.

```
shell > ./test_read.py -p /dev/ttyUSB1 -b 57600
^CEnding
Summary: count=49920 , failed=0
```

The number of bytes counted should match those displayed



UART Loopback

Baud rate	Stop(bit)	Data(bit)	Parity	Flow
57600	1	8	None	None

Bytes Read: 57120 Failures: 0

Bytes Written: 49920

Figure 5: Completing the Write Test

Changing the Baud rate

You can also change the Baud rate to other settings and perform these tests. The only issue is that if you want to change Baud rates you must first disconnect the USB cable, plug it back in and then follow the instructions from configuring baud rate section.