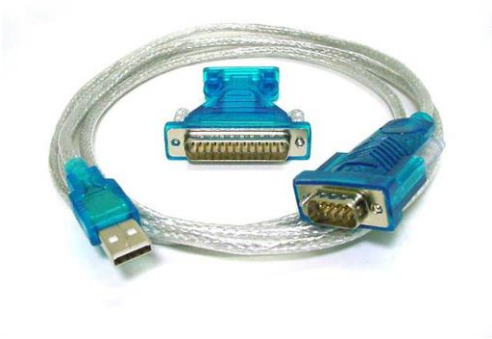
Setup Hardware and Software

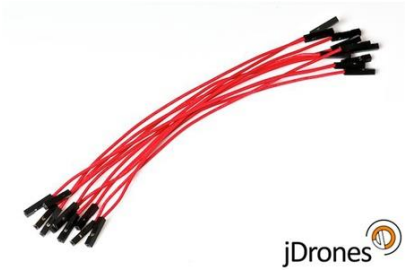
Hardware requirement

- One USB to rs232 (or usb to com)as link below: http://www.monoprice.com/Product?p_id=2067&gclid=Cj0KEQjw6O-9BRDjhYXH2bOb8Z4BEiQAWRduk5vVwmZBndtXIDwF_VSiAnriBhxEvaLAqJS2TSDiZpYaAli98P8HAQ



- One jumper cable to connect pin 2 and pin 3 of USB to COM for echo as link below:

http://store.jdrones.com/jumper_cable_10cm_red_p/cbl11p10red.htm



Download Required Software

- Java8 JRE such as JavaSetup8u101 software for Windows10 64bit. you can download at link below: http://java.com/en/download/win10.jsp
- A ch341_hl-340_windows_drivers file for usb to rs232 or PL2303_Prolific_DriverInstaller_v1_14_0.zip file. Depend on usb to com you use.

Links for download:

http://www.cesareriva.com/usb-to-serial-converter-hl-340-drivers/

http://www.prolific.com.tw/US/ShowProduct.aspx?p_id=225&pcid=41

For MAC:

OS X 10.11 El Capitan: http://www.wch.cn/download/CH341SER_MAC_ZIP.html

OS X 10.12 Sierra https://tzapu.com/ch340-ch341-serial-adapters-macos-sierra

Install software

Needed software

1) RXTX Java serial port driver(http://rxtx.qbang.org/pub/rxtx/rxtx-2.1-7-bins-r2.zip)

For MAC, http://jlog.org/rxtx-mac.html

2)Java TestDriver https://visualthreat-my.sharepoint.com/personal/jeffrey_visualthreat_com/_layouts/15/guestaccess.aspx?guestaccesstoken=Qsz4XCEpA8Cwzcco9cb2yuXwLodYo0Q1g%2bcffrjRxrY%3d&docid=0d71c4682768c4225aa318cb5026c585e&rev=1

Steps

- Install java:

Please double click on JavaSetup8u101 file to install and then click next.

- Install driver for USB To COM:

After download extract file and then double click on PL2303_Prolific_DriverInstaller_v1_14_0 to install.

- Install RxTxcomm for java:

- Copy and paste RXTXcomm.jar into \jre\lib\ext folder.(E.g: C:\Program Files (x86)\Java\jre1.8.0_101\lib)

- Copy and paste rxtxSerial.dll into \jre\bin folder. (E.g: C:\Program Files (x86)\Java\jre1.8.0_101\bin)

Running TBOX_driver_java

- Copy two files driver-1.0-SNAPSHOT-allinone and conf.properties to your computer.
- Open Command Prompt and then go to folder containing the two files above

- Modify conf.properties **deviceId** to a name you like

Run command below:

java -cp ./driver-1.0-SNAPSHOT-allinone.jar com.visualthreat.testing.TestDriver -p ***COM5*** -c C:\Work_2016\TMP\tbox_driver_java\conf.properties

Note: You must change COM port to the port you have and the full path file to the conf.properties file. (C:\Work_2016\TMP\tbox_driver_java\conf.properties => <Your Path>\conf.properties>)

You will see something like below for a successful run:

```
12:22:18.538 [main] INFO com.visualthreat.testing.TestDriver - TestDriver Starts

Stable Library

=========================================

Native lib Version = RXTX-2.1-7

Java lib Version   = RXTX-2.1-7

12:22:18.614 [main] INFO com.visualthreat.testing.TestDriver - id=0 connection on COM5 established

12:22:18.617 [Thread-3] INFO com.visualthreat.testing.TestDriverSocketWorker - Connecting to 114.55.62.8:18881

12:22:18.925 [Thread-3] INFO com.visualthreat.testing.TestDriverSocketWorker - Authenticating myself...

12:22:18.994 [Thread-3] INFO com.visualthreat.testing.TestDriverSocketWorker - Waiting for auth response...

12:22:19.244 [Thread-3] INFO com.visualthreat.testing.TestDriverSocketWorker - Successful auth, session: 5eb9b784-3596-4dd8-9207-6632f2072759, seqNum: 0

12:22:20.196 [Thread-3] INFO com.visualthreat.testing.TestDriverSocketWorker - Received GetDeviceInfo
```

Go to web site https://staging.visualthreat.cn and login

To send command to TBox_driver_java please through steps below after login:

- Select Testing Scenarios
- Select Commands Tab
- Select device (E.g: tbox3_test_duy, the name you choose above). You can change device id in conf.properties file.
- Select "Execute Test" command

Typing {"type":"P", "count": 1, "id":1872, "dlc":8, "data":[0x40, 0x05, 0x30, 0x11, 0x0, 0x0, 0x80,0]} in the Payload textbox(you can type any CAN command)

Click Send button. You'll see log events like below in response "Message" Text box. Basically TestDriver echo back the command you send from Cloud because the serial port is setup in echo mode.

```
Session ID: 5eb9b784-3596-4dd8-9207-6632f2072759

Server received log num: 1
```

Server received CAN Message:id=1872;dlc=8;data=[64, 5, 48, 17, 0, 0, -128, 0]

Running in local log collection Mode

When you specify configuration "localLogFolder" in conf.properties file, TestDriver will run in local log collection mode without talking to Cloud.

localLogFolder=/var/log/services/local

If you want to running in Cloud mode, you need either comment the following line out as following or just remove it.

#localLogFolder=/var/log/services/local

When TestDriver runs in local mode, you will see the following log line in console output:

TestDriver is running in local log collecting mode

When TestDriver starts in local log collection mode, it automatically collects log until it gets killed. The log will be saved into folder specified by "localLogFolder" in a log file whose name is like "local_<unix epoch timestamp>.log"

Running Java Tests

Prepare Environment

1) Download Java tests package at https://visualthreat-my.sharepoint.com/personal/jeffrey_visualthreat_com/_layouts/15/guestaccess.aspx?guestaccesstoken=D86itUm4yU3FT%2boqy%2fp1s9JjspHwZU0yLgT%2fZv%2bSLjs%3d&docid=0bb15a97327b54e1690b5e2d9fd59c0e2&rev=1

2) Create folder c:\tests\log and c:\tests\state

3) Copy tests-allinone.jar, logback.win.xml and other files in the zip file into folder c:\tests

4) cd c:\tests

See help please run the following command:

java -jar tests-allinone.jar -h

A sample command(on Windows) to only run ECU Ids and Service discovery not sub function discovery.

*java -Dlogback.configurationFile=c:\\tests*logback.win.xml *-jar tests-allinone.jar -t UDSServiceDiscoveryTest -d COM7 -b 125000-rw 400 -s c:\\tests\\state -a "#e services"*

If you want to run only SubFunction discovery, you can specify

-a "#s subfunctions"

You'll see log files(tests.log and traffic-*.log) are created under c:\tests\log folder*

Supported Tests are as following and all tests are depending on UDS Discovery Tests completed the first two steps: ECU ID and Service Discovery

UDSServiceDiscoveryTest

SecurityAccessTest

CANFuzzTest

WriteDataByIdentifierTest

WriteMemoryAddressTest

ReadDataByIdentifierTest

RoutineControlTest

FuzzSubFunctionsTest

SimulateReprogrammingTest