

Android Installation

Revision 1.0 - 8th June 2015
www.touch-base.com/documentation/installation



[Deliverables](#)

[Requirements](#)

[Installation](#)

[Uninstall](#)

[Supported programs](#)

[Limitations](#)

[Contact](#)

Welcome to UPDD Android platform specific installation instructions and related notes for UPDD version 5.2.x.

UPDD provides a touch solution for Android systems. This implementation supports the Android operating system on ARM7 hardware.

This may be provided for other versions of Android and target processors if there is a demand for this.

Functionality is broadly similar to the UPDD desktop variants; with the following main variations.

- 1) There are no GUI based console and test utilities. The UPDD [command line utility](#) is however included as a means to adjust the software settings.
- 2) Extension features provided by the [UPDD Daemon task](#) are not available. Functions supported by this UPDD module will be added as and when required.
- 3) Only USB devices are supported.

These notes should be followed to install the UPDD pointer device driver on Android platforms:

Driver build history

Release	Date	Change
5.2.xxxx	June 2015	Initial release

License Notice

The software is licensed software and as such requires a license per system when the production version of the software is installed. Production software is either supplied by pointer device manufacturers or system integrators (who are entitled to distribute the driver) or is available directly from [Touch-Base sales](#).

Evaluation version restrictions

Evaluation software, which can be supplied from Touch-Base, is mainly used to test the touch function is working and has certain restrictions. After 100 touches or 24 hours use the touch screen will stop working.

Deliverables

The software will be delivered in a tgz file named android_armeabi-v7a_05_02_NNNN.tgz; where NNNN is the UPDD software build number and is normally downloaded from a HTTP download link as sent in an email.

System Requirements

Component	Note
Kernel 4.x	The software was initially developed and tested on Android version 4.2 Jelly Bean, API release 20.
ARM7	This software is currently only available for ARM7 or compatible processors and has been tested on: MiiPC Android 4.2 Jelly Bean CuBox-i4 Android 4.4 KitKat
Touch screen	USB touch device
Root access	Given that the driver must be embedded you will need root access to the device. In the Android world this is referred to as the device being "rooted". You will need to consult your device documentation or other information available online. This article provides a useful general guide to rooting your device http://www.cnet.com/uk/how-to/how-to-root-your-android/ .
Terminal emulator	You will need access to a terminal emulator. Typically you will have installed one as part of the rooting process. If not this one https://play.google.com/store/apps/details?id=jackpal.androidterm&hl=en is the utility we have used.
Access to TCP IP port 4141	The driver requires access to TCP IP port 4141 for internal computer processing only.

Technical considerations

The touch interface utilises the uinput interface. uinput is a kernel module that allows access to the input subsystem from user mode. Unlike UPDD drivers for other Linux variants this is the only mode supported.

TCP/IP port 4141 is used for local IPC (e.g. between the UPDD calibration program tbcilib and the driver).

The software is implemented using the NDK, i.e. with native code and is therefore bound to a specific architecture. In theory later versions of Android on the same processor architecture should be supported; but this is subject to verification.

Software delivery

In this environment, UPDD is supplied as a number of separate components. Touch-Base utilises virus detection software on all of our systems but recipients of the software should pass the files through their own virus checking software before proceeding with installation.

Normally a software package will be supplied with just one touch controller supported in which case the settings file can be used as supplied.

Components

The UPDD Android package consists of the following components and indicates their location after install:

File Name	Description	Installed to (by standard script)
tbupddwu	Touch driver	/system/vendor/updd
tbutils	Utility program	/system/vendor/updd
tbupdd.ini	Driver settings	/data/data/updd
install.sh	Standard install script	Not installed
startupdd	set environment for and run driver	/system/vendor/updd
upddenv	Helper script to execute updd tools	/system/vendor/updd
liblibACE.so	3 rd party lib for IPC and synchronization	/system/vendor/updd/lib
liblibgnustl_shared.so	C++ standard template library	/system/vendor/updd/lib
liblibusb1.0.so	USB interface	/system/vendor/updd/lib
liblibQt5Core.so	Qt core functions libray	/system/vendor/updd/lib
liblibc.so	C run time	/system/vendor/updd/lib
liblibdl.so	Android system component	/system/vendor/updd/lib
libliblog.so	Android system component	/system/vendor/updd/lib
liblibm.so	Android system component	/system/vendor/updd/lib
liblibstdc++.so	C++ run time	/system/vendor/updd/lib
liblibz.so	Compression library	/system/vendor/updd/lib
apk\tbcalib.apk	UPDD calibration app installer	Determined by Android

Installation

The software is delivered in a tgz file named android_armeabi-v7a_05_02_NNNN.tgz; where NNNN is the UPDD software build number.

The following is a guide to installing the software; expert users can adapt this procedure as required, for example to install the software in different locations.

Note: the install script provided performs certain steps based on typical Android devices. Given the potential risk of executing incorrect commands as a root user you should check that the steps this scripts takes; which are detailed in a later section are valid for you device

If you are unsure you should review the script and execute the commands contained one at a time.

The example process is as follows.

- 1) Copy the delivered file to a USB stick.
- 2) Plug the stick into the Android device.
- 3) Open your terminal emulator.
- 4) Enter the command "su" to gain superuser privileges. NB you will have needed to "root" your device to allow this. Also note that it is important to take care when working as superuser as you can damage the system with incorrect commands.
- 5) Navigate to the usb drive using the command

```
cd /mnt/media/<drive name>
```

note that if you type "cd /mnt/media/" then hit the TAB key Android will list the available options, typically pick one starting USB. If you type the first few letters of the name and hit TAB again Android will complete this for you.

- 6) Extract the software using the command
- 7) After noting the cautionary notes above execute the install script using the command

```
./install.sh
```

The install script performs the following actions.

- 1) Sets the file system /system to be writable.
- 2) Creates the folder /system/vendor/updd
- 3) Copies the driver and related files to /system/vendor/updd
- 4) Creates the folder /system/vendor/updd/lib
- 5) Copies the associated library files to /system/vendor/updd/lib
- 6) Creates the folder /data/data/updd
- 7) Copies the driver settings to /data/data/updd/tbupdd.ini
- 8) Creates a backup of /system/etc/init.rc as /system/etc/init.rc.preupdd
- 9) Adds an entry to /system/etc/init.rc to launch the startupdd script
- 10) Uses the command "tbutils fixini" to prepare the settings file for use (adds a checksum to the file).

- 8) It might be useful to refer to the diagnostics section at this point to check that all embedded components are working as expected.
- 9) Restart the system.
- 10) In your Android file viewer navigate to the USB drive and click tbcilib.apk and follow the prompts to install the calibration program.

Notes about script upddenv

In this document there is reference to script upddenv which is used to invoke UPDD utilities, specifically the command line utility tbutils. This script is use because UPDD components use shared libraries. Given these libs might conflict with files shared with other applications a private copy is used and these are not stored in the system library path (\$LD_LIBRARY_PATH). The upddenv script prepares the environment for the current session only to include the path for such libraries. This allows an easy way to use UPDD Linux command line tools, especially tbutils.

Uninstall

To uninstall the software you need to manually reverse the installation procedure as follows:

There is no single uninstall utility; this is because Android provides a system to manage apps; and this should be used to uninstall the calibration app.

To uninstall the driver itself open the terminal emulator and enter these commands

```
su
mount -o remount,rw /system
```

```
rm -rf /system/vendor/updd
rm -rf /data/data/updd
```

the following step is optional, see note below

```
cd /system/etc
cp init.rc.preupdd init.rc
```

```
mount -o remount,ro /system
```

Note, only execute the command "cp init.rc.preupdd init.rc" if you are sure no changes have been made to init.rc since the driver was installed. Otherwise use an editor to remove the updd entries, or simply leave them in place; for most practical purposes this will do no harm.

Using the supported programs

Utilities

Program	Description
tbcilib	To calibrate the system simply open the app UPDD Calibrate from your apps list. A screen with targets (usually crosses) is shown. Touch each until a blue circle appears. When all points have been touched the calibration is complete, the app terminates and you can use touch.
tbutils	The UPDD command line utility is included as a means to adjust the software settings and list driver and controller configuration information. cd /system/vendor/updd ./upddenv tbutils [parameter]

Calibration

Calibration is a procedure used to align the touch screen with the graphical display area or desktop segment. When using mouse emulation interface the mouse cursor should normally position itself under the point of contact. If this is not the case then calibration will be required.

Command Line Utility

This utility is described in a [separate document](#) and can be used to list driver and system information and retrieve / change driver settings. The utility is located in folder /system/vendor/updd.

The driver is custom built for specific touch device and will only work if the device is located on the system. For USB the device is located via the USB vendor and product id. The 'tbutils list' command will indicate if a device has been located.

Should the above command not list any devices then you can use the dmesg command "dmesg | grep usbif" to list USB devices within the system along with their vendor and product ids. The vendor and product id must match that configured for the touch device in the UPDD driver.

Other commands can be used to show the device is working , such as the 'tbutils caltouches' to show calibrated touch data received from the controller:

A user is free to create a desktop shortcut to any of these utilities. As the method to do this depends on the desktop manager in use, please refer to the help for your desktop manager.

Diagnostics / troubleshooting

This section lists a few useful commands to check everything is working as required or identify the problem area.

Running the driver.

Only do this if you have run the installation steps above and NOT restarted the system or you suspect the driver is not running. After a restart the driver should be working.

```
cd /system/vendor/updd
./startupdd
```

The terminal emulator will now be running the driver and will not respond. You should open another terminal emulator window to enter other commands.

Checking that the driver is activated and the API programs are able to communicate with the driver.

```
cd /system/vendor/updd
./upddenv tbutils reload
```

If all is okay you should see no response and the terminal emulator should return to the prompt.

You might see an error at this point.

For example "unable to open connection to driver suggests that the driver is not running. If you have run the driver manually as above recheck you did this correctly.

If the system has been restarted the driver is loaded by an entry in /etc/init.rc; you should start by checking this.

A message "no valid settings found" suggests that the file /data/updd/tbupdd.ini cannot be found in the expected location or that it is invalid.

Checking the touch device is recognised.

Enter the commands

```
cd /system/vendor/updd
./upddenv tbutils list
```

If all is okay you will see your touch device listed.
If not you could try

```
cd /system/vendor/updd
./upddenv tbutils listcontrollers
```

this lists the device types supported by this package.

Diagnostic tracing

If you are having no luck and you need to contact Touch-Base support you will need to follow this process

Note: this assumes that nothing else has modified the file init.rc since UPDD was installed. If this is not the case then you must manually remove the UPDD entries from init.rc

- 1) execute the commands


```
cd /system/vendor/updd
./upddenv tbutils global setting dw debuglevel 5
```

 If this command fails you will have to edit tbupdd.ini manually later in this process
- 2) Reinstate init.rc


```
su
mount o remount,rw /system
cd /system/etc
cp init.rc.preupdd init.rc
mount-o remount,ro /system
then restart.
```
- 3) Only if step 1 failed
 use any system editor to change the setting debuglevel to 5 in the file /data/data/updd/tbupdd.ini
- 4) Start the driver in debug mode


```
cd /system/vendor/updd.
./startupdd >
```

 Information is then displayed in the terminal window. Please capture this and send it along with tbupdd.ini to Touch-base support

Current Limitations

UPDD was originally developed for Windows and has since been ported to other OS. When porting to other OS most key updd functionality is supported except where the functionality is OS specific or it is noted in the exclusions below.

The driver uses a background daemon process, [aidaemon](#), to implement some features and monitoring functionality, such as a system tray type menu, monitoring screen rotation etc. Aidaemon is NOT included in the current Android build and therefore any of the supported functions required in an Android environment will need to be implemented on a per request basis.

Mouse Emulation

This driver supports desktop mouse emulation to allow touch interaction with the desktop and applications. Applications can also utilise the [UPDD API](#) to receive touch data directly into the application.

Exclusions

As discussed above, functions associated with the driver's daemon process are not support under this OS.

Further, these restrictions apply:

- There are no GUI based console and test utilities.
- Only USB interface is supported, custom built for a specific device as required.
- Right click not currently supported – can easily be added if required.
- Interactive touch – visual notification of right click count down
- Dynamic detection of system language and no localisation (English only)
- Multi-monitor support untested
- No sound on touch or during calibration

Contact

For further information or technical assistance please email the technical support team at technical@touch-base.com