TOUCH BASE documentation

UPDD TUIO Server

Revision 1.11, 15th Sept 2014
www.touch-base.com\documentation\general

Overview Mac OS X Windows Linux Calibration Uninstall Testing Interfaces Contact

Overview

TUIO (Tangible User Interface Object) is an open protocol for the communication of data from devices like a multi-touch display, an interactive surface, or a computer vision-based motion tracker. The sorts of information it describes includes touch events and object locations, much like with the Microsoft Surface, as per the video here. TUIO uses a client/server model for communicating this data, so any program that is a TUIO client can receive touch or object data from any TUIO server.

TUIO is based on Open Sound Control (OSC) a specification for sending messages interactively, which while originally intended for sharing music and performance data, has been increasingly used for other kinds of non-musical data.

OSC transmits information across standard IP networks and the internet using User Datagram Protocol (UDP – see interfaces), so it's supported by a lot of systems, including Windows, Mac OS X, and Linux. Any system that supports OSC can also support TUIO, and consequently, there are many TUIO client and server programs across these platforms. Further, because TUIO uses UDP for communication, a TUIO server and client needn't be running on the same computer, or even on the same local network, though both running on the same computer probably covers most usage cases.

The UPDD TUIO server program implements a TUIO server such that any application that is a TUIO client and receives TUIO touch data can be controlled with a UPDD driver supported touch-screen, thus expanding the number of multi-touch enabled applications that UPDD can support, the schematic being:

Touch hardware > UPDD driver > UPDD TUIO Server > TUIO client application > End user interaction

The main limitation is that UPDD TUIO Server does not support object data, as the UPDD only tracks touches on a multi-touch display. However, the main use of TUIO is to communicate multi-touch data, so most TUIO applications will be usable with the UPDD in spite of this limitation. If it should become necessary for the UPDD to support TUIO objects as well, an interface could be created to allow for the creation and positioning of virtual objects, much like the TUIO simulators that exist for testing and demonstration purposes.

Currently the UPDD TUIO Server exists as a stand-alone application that needs to be compiled for each operating system but will in future be integrated into the main driver so will become part of the standard driver offering for each OS.

Configuration / Settings

All the settings are held in the UPDD Settings file, general driver settings section and can be adjusted with the TUIO setting dialog documented below or using one of the alternative update methods.

Given that TUIO servers communicates using UDP then the only TUIO specific configuration necessary for the TUIO server is the IP address, port number and message packet size. Since most usage cases for a TUIO server is to have it running locally on the same computer that the TUIO client programs are running on, a reasonable default configuration is for it to use localhost with the default TUIO IP address (127.0.0.1), port number (3333) and packet size.

The UPDD TUIO Server utility accepts command line arguments to override the TUIO settings and define settings for the ***current TUIO session only*** (the setting file entries stay intact):

-b	bind-address (Specify bind-address)
-p	port (Specify port)
If either of the above arguments are missing the information is retrieved from tbupdd.ini if found otherwise uses defaults.	
-i or --no-icon	option to run without an icon (i.e. as a daemon). – Linux and Mac only
-h or –help	help (Displays usage text)
And since May 2012	Mouse Emulation settings
-m	With this version UPDD mouse emulation is on by default. This option can be used to enable / disable mouse emulation. 0 = off/disabled, 1 = on/enabled.
-n or --no-mouse-in-apps	If Mouse Emulation is enabled all applications will receive mouse emulation clicks. Applications receiving touches via TUIO interface may not want to also receive mouse emulation clicks and this option is used to specify one or more process names that should not receive mouse emulation clicks. Whenever a process matching one of the names is 'front most' mouse emulation will be automatically disabled. It's not case sensitive and under Windows it's not necessary to include the ".exe". Executables with spaces in their name should be enclosed in quotes. Windows example: To run UPDD TUIO server with mouse emulation off in both QtTuioPaintDemo.exe and an executable named Touch App.exe, you would invoke the service with the command line: **UPDD TUIO.exe -n QtTuioPaintDemo "Touch App"** Once a window from either of those apps becomes 'front most', mouse emulation will automatically turn itself off, regardless of the setting in the system tray menu. Switching to a different process will reset mouse emulation to the menu setting.
Since April 2013	TUIO timestamp options
-r or --use-updd-rate	The TUIO server now timestamps TUIO frames. By default this is as close as possible to 60 frames per second as per the TUIO specification. This setting can be used to transmit the frames at the same rate they are received from the driver albeit this does not conform to the TUIO

specification: 0 or off = 60 FPS, 1 or on = UPDD driver rate. Any application written to retrieve frame times should now receive valid timestamps. E.g.http://liblo.sourceforge.net/ lo_message_get_timestamp API call.

In our tests we achieved a TUIO data rate up to about 200 packets per second, although this speed will vary depending on hardware in use.

Since Dec 2012 a TUIO settings dialog can be invoked from the system tray or menu bar icon to adjust the settings as required.

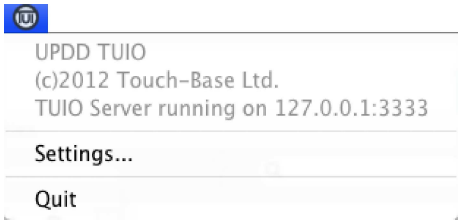
System Tray or Menu bar icon

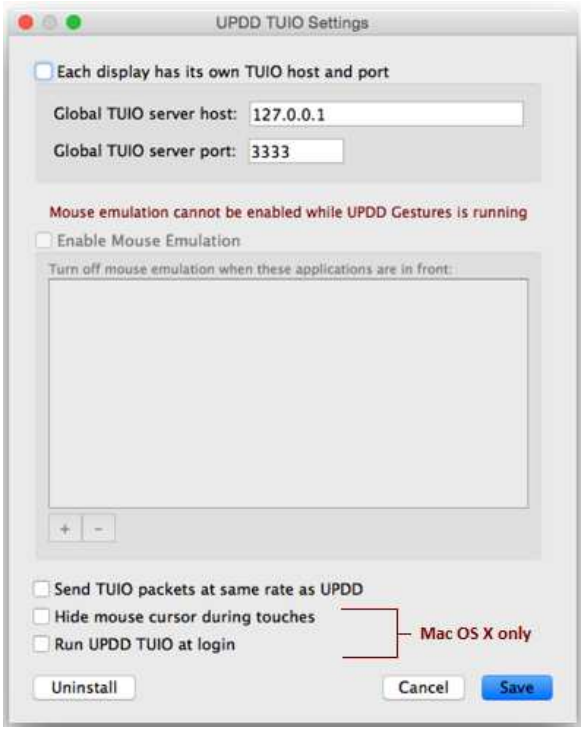
When the UPDD TUIO server loads it will install a system tray or menu bar icon as follows:

Clicking on the icon reveals configuration information about the UPDD TUIO server and offers options to adjust the settings and to Quit the TUIO service.

Note: if using Ubuntu, you may need to follow these directions to get the UPDD TUIO system tray icon to appear:

http://techpublic.net/linux/show-selected-apps-in-ubuntu-unitys-system-tray-with-dconf-editor-how-to/





Notes:

1. If the TUIO server software is invoked using one or more command line parameters, they will override the settings defined in the UPDD settings file for that invocation of TUIO server. The settings in the file are not changed but are temporarily overridden. In this case, when viewing the settings via the settings dialog whichever settings are overridden will be grayed out, displaying the overridden value.

2. If running with mouse emulation enabled (able to use the touch screen outside of the TUIO applications) then the TUIO applications will also receive mouse emulation clicks. To exclude any application from receiving mouse emulation clicks define the names of the TUIO client applications or use the –n command line parameter when invoking the TUIO service.

3. Under Mac, if running the gesture software then the Enable Mouse Emulation option is disable as mouse emulation must be disabled if gestures are running.

4. When upgrading or reinstating the TUIO software the software will utilise any defined settings held in the settings file from a previous utilisation of the TUIO server software.

5. 'Hide mouse cursor during touches' and 'Run UPDD TUIO at login' are current only available in Mac OS X.

6. As of Jan 2014 now supports multiple TUIO interfaces. You can now configure UPDD TUIO to send touches to different ports depending on which touch screen it comes from, in essence running multiple TUIO servers at once, one for each touch screen. There is now a checkbox that switches between using the same TUIO host and port for all displays, and using different settings for each display. When checked, it lists each display configured in the driver and held in the settings file (tbupdd.ini) and gives a field for setting its host and port.

   Notes: There's no command line options for the new settings. The settings window doesn't resize itself properly when switching > between using one TUIO host/port and using different ones for each monitor.

The settings above would result in the following TUIO related settings in the settings file:

```
tuio apps without mouse emulation=QtTuioPaintDemo
tuio hide cursor=true
tuio host=127.0.0.1
tuio mouse emulation=false
tuio port=0x00000D05
```

The software is compiled under 32 bit except if using UPDD version 5.0.x 64 bit version in Windows in which case you must run the Windows 64bit version. Since June 2013 both 32 and 64 bit Linux versions are available.

Invoking TUIO Client application program considerations
If the TUIO Bridge software is invoked requesting that normal touch mouse emulation is disabled, to avoid the TUIO applications receiving touch from two interfaces, then the desktop will no longer react to touches. For this reason it is suggested that named client apps to be defined rather than disabling mouse emulation for the whole system.

However, if you must run with the system wide mouse emulation disabled and touch is used to invoke the bridge software along with the TUIO client application then some form of batch file or script be utilized that invokes both the TUIO bridge and TUIO client application should be employed.

For example, under Windows you can use a VBScript file to invoke the applications, as per this example:

```
Tuio_bridge.vbs – file name
wScript.Echo "This is a test Script"
set objshell = CreateObject("wScript.Shell")
objShell.CurrentDirectory = "C:\Program Files\UPDD\"
objshell.run """C:\Program Files\UPDD\UPDD TUIO.exe"""
objShell.CurrentDirectory = "C:\Program Files\Application Folder\"
objshell.run """C:\Program Files\Application Folder\TUIOClientApp.exe"""
wScript.Quit
```

OS Implementations

Currently UPDD TUIO server exists as a stand-alone application that needs to be run for each operating system but will shortly be integrated into the main driver so will become part of the standard driver offering for each OS. The software is compatible with UPDD 4.1.10 or above.

Mac OS X

The UPDD TUIO server for the Mac OS X is available here, based on the UPDD version 5.0.2 or 5.1.x. Some UPDD driver installs also install the TUIO software automatically. If the TUIO software is installed as part of the UPDD driver install it will be installed in /Applications/Utilities along with the other UPDD utilities. If manually installing the software it is recommended that the software is placed in this location so that is it automatically deleted by the UPDD driver uninstall program if and when UPDD is uninstalled. The uninstall will also remove any start up item for the TUIO server.

To utilise this service in a Mac environment you need to simply run the 'UPDD TUIO.app' application with UPDD 4.1.10 or above installed.

The TUIO server application interacts with the core UPDD driver to receive all touches and then post TUIO events. To Quit the service use the Quit option on the menu bar icon or your can safely quit using the kill command, or using the Activity Monitor.

Invoke at system startup
It may be desirable to have the TUIO Server running at all times. In this case you need to create a startup item for the TUIO server. Since TUIO server software issued Dec 2012 there is an option in the TUIO server settings dialog to create the start up item. If the software is installed as part of the UPDD driver installation the start up item is added during the install.

Implement browser TUIO touches
We have been asked on a few occasions how you might utilise multi-touch gestures in a browser under Mac OS X. We are aware that when using UPDD Gesture software the gestures cannot be used to control touch-enabled web pages but are performed on the browser dialog.

The problem is that OS X doesn't pass its system-level touch events to web browsers as HTML5 / javascript touches. So while these pages will work with an iPad (and similar) it can't work in OS X, even when using Apple's magic trackpad.

However, extending some work on browser multi-touch (and the resultant magictouch.js developed by Boris Smus) we have been able to implement a 'hack' such that all web pages are slightly modified so that touches will work. This 'hack' converts TUIO touches to HTML5 touches and we have solutions for Chrome and Firefox as documented below.

Before setting this up in either Firefox or Chrome, it's necessary to first install the npTuioClient plug-in. Download link and installation instructions: https://github.com/fajran/npTuioClient. In some of our tests we had issues with this plugin and used the latest one at https://github.com/fajran/npTuioClient/issues/8 and we recommend this one is used on most recent Mac OS X versions (we tested on 10.6.1 and 10.8.4)

Essentially all that's necessary is to download npTuioClient.bundle (the older one creates a file called npTuioClient.plugin, the new one a file npTuioClient.bundle) and copy it into "~/Library/Internet Plug-ins". This plug-in will be available for all browsers. We recommend quitting any open browsers before installing it.

Instructions for Chrome:

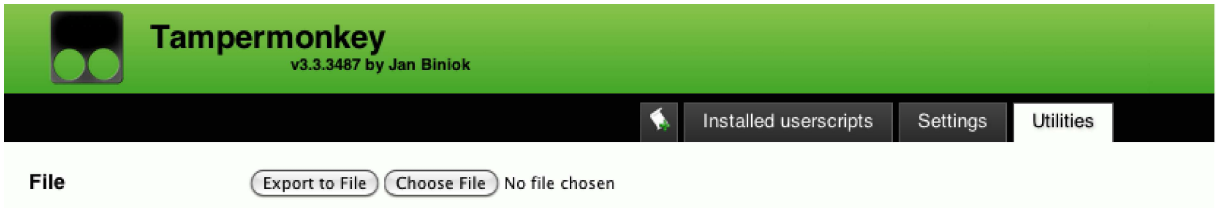
1. Launch Chrome, go to this URL to install the extension Tampermonkey:
https://chrome.google.com/webstore/detail/tampermonkey/dhdgffkkebhmkfjojejmpbldmpobfkfo

2. Add Tampermonkey to Chrome

3. A button will appear in the upper right portion of the window that looks like a box with two dots on the bottom of it. Click this and pick "Dashboard":

4. Click the "Utilities" tab in the Tampermonkey settings.

5. Click "Choose File" (download first from link below)



6. Select "tuioTampermonkeyScript.txt", Click OK in the dialog that appears to continue.

Instructions for Firefox:

1. Launch Firefox, go to this URL to install the extension Greasemonkey:
https://addons.mozilla.org/en-US/firefox/addon/greasemonkey/

2. Add it to Firefox

3. Restart Firefox

4. Drag the file "addtuio.user.js" (click to download) on to the Firefox window. It will prompt you to install a Greasemonkey script, click 'Install'.

After running these instructions either browser should start working with TUIO touches. If UPDD TUIO is running, UPDD touches should be received by the web browser. Here's a few web pages we've been using for testing:

http://stickmanventures.com/labs/demo/onscreen-keyboard-html5-canvas-multitouch/
http://tomicloud.com/2012/03/multi-touch-demo
http://dev.globis.ethz.ch/jqmultitouch/1.0/behaviours.html

This 'hack' does have some limitations:

- Webpages that try to detect if a browser supports multi-touch may not work, depending on how they go about doing that. (There are several methods, some will fail, others will work.)
- For the touches to properly line up with a web page, it's necessary for the browser to be running in full screen mode, without toolbars. Chrome's "presentation mode" accomplishes this. It may be possible to remove this limitation if required.
- The touch events this method produces may not be exactly the same as touches on an iPad or Android device, so some webpages may not work as expected.

If you have a specific webpage that does not work with this hack we may be able to improve the scripts to suit your requirements. Please contact us for further assistance.

This hack is used at your own risk as we cannot guarantee that there may be some browser issues that arise from implementing these extensions and TUIO scripts. Any feed back much appreciated.

Windows

The UPDD-to-TUIO server for the Windows:
UPDD 4.1.10 (32bit)
UPDD 5.0.2 32 bit is available here or 64 bit is available here.
UPDD 5.1.x 32 bit is available here or 64 bit is available here.

To utilise this service in a Windows environment you need to simply run the 'UPDD TUIO.exe' application with UPDD 4.1.10 or above installed.

Copy the UPDD TUIO.exe to the /program files/updd folder and run as appropriate. If run from a different folder you will need to copy the tbapi.dll, ace_UPDD_5.6.2.dll into the same folder.

The TUIO server application interacts with the core UPDD driver to receive all touches and then post TUIO events. Use the System Tray Quit option to stop the service.

You may be required to install Microsoft Visual C++ Redistributable Package. The application will report a required .DLL (MSVCP100.dll) is missing if the run-time is not installed.

Invoke at system startup
It may be desirable to have the TUIO Server running at all times. In this case you need to create a short cut in the start up folder to the TUIO server

Linux

The UPDD TUIO server for the Linux
UPDD 5.0.2 32 bit is available here or Linux 64 bit is available here.
UPDD 5.1.x 32 bit is available here or Linux 64 bit is available here.

To utilise this service in a Linux environment you need to simply run the UPDD TUIO application with UPDD 5.0.x or above installed.

The instructions for using this under Linux is as follows:

cd /opt/tbupddlx
cp <downloaded file> .
sudo tar –xzvf UPDD-TUIO.tgz

To run
export LD_LIBRARY_PATH=/opt/tbupddlx:$LD_LIBRARY_PATH
"/opt/tbupddlx/UPDD TUIO"

If running form a terminal Window use Ctrl Z to terminate the process.

These two lines can be added to an appropriate startup script if required.

The TUIO server application interacts with the core UPDD driver to receive all touches and then post TUIO events.

Uninstall

If the TUIO settings dialog shows an Uninstall option select this to uninstall the software.

Alternatively, use the UPDD TUIO settings dialog to disable any start up entry and then simply locate the TUIO application file and delete as appropriate to the host operating system. You may also wish to delete the TUIO settings in the settings files although this is not really necessary as they will not be utilised until the UPDD TUIO server is reinstated.

Alternatively, uninstalling the UPDD driver will also remove the software if under Windows or Linux it is located in the UPDD Application folder or under Mac OS X it is located in /Applications/Utilities. The driver will also attempt to remove any start up item associated with the TUIO server.

Testing

Once the TUIO server application is running then any application that is a TUIO client should respond to touches.

We used the following TUIO applications for our tests:

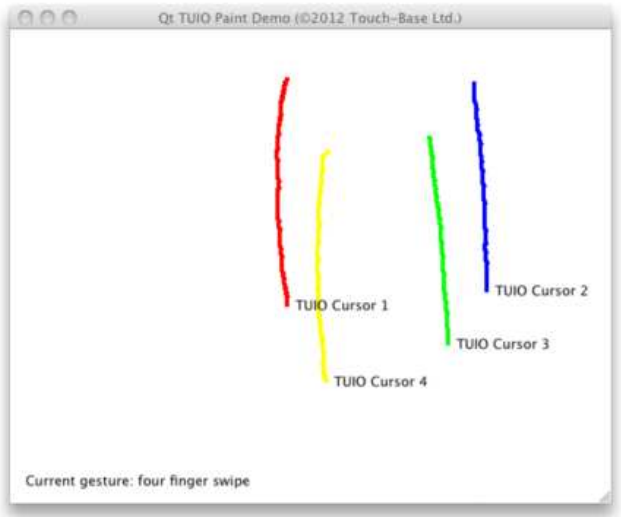
TUIODemo

For the basic Java TUIO client:
1. Go to http://www.tuio.org/?software
2. Click the link under "TUIO Client Reference Implementations" named TUIO_JAVA.zip. TUIO_JAVA.zip should download automatically
3. Unzip the file
4.In the TUIO_JAVA folder, open TuioDemo.jar. A window should appear with the title "TuioDemo" which should display touches.

Java must be installed on a system for the above demo to work
http://www.java.com/en/download/index.jsp

QtTUIOTouchDemo

This demo program was written in-house to test our UPDD TUIO server and is available here. In this example the TUIO server has been loaded and the QtTUIOTouchDemo program listed to turn off mouse emulation when top most so that only the TUIO touches are recorded: The calculated gesture is also shown. An example of 10 touches can be seen here.



Calibration

The TUIO spec states "The TUIO coordinate system is normalized for each axis and is represented by floating point numbers in the range from 0.0f to 1.0f. In order to compute the X and Y coordinates for the 2D profiles a TUIO tracker implementation needs to divide these values by the actual sensor dimension, while a TUIO client implementation consequently can scale these values back to the actual screen dimension. "

Based on the above it is our understanding that it is the responsibility of the client application to provide the calibration scaling. It is a limitation of some of the example programs not to account for the size and position of the application window and the entire dimension of the physical touch screen is scaled into the application window. In this case a real world application designed for touch (as opposed to an off screen input such as a non-interactive whiteboard, track pad etc) would need to run full screen or track its size and position and scale accordingly.

Interfaces

The common interface for TUIO is UDP. However it is our understanding that two additional interfaces were introduced for using TUIO with Flash, initially TUIO/FLC followed by TUIO/TCP. The reason was, until recently, that Flash didn't support either TCP or UDP sockets, so TUIO/FLC was created. It uses a communication protocol from Flash called "LocalConnection" to send touch information directly into a Flash applet. Flash did eventually add support for TCP sockets, but apparently they're not so good for TUIO due to higher latency.

UPDD-TUIO-Bridge currently only supports UDP (no TCP or FLC) and we could add support for other interfaces. However, we do not believe this is necessary for two reasons:

- There is software that can bridge between them. In particular:
  http://qkaindl.com/software/udp-flashlc-bridge - Bridges TUIO/UDP and TUIO/FLC

  http://code.google.com/p/udp-tcp-bridge/ - Bridges TUIO/UPD and TUIO/TCP.
  This program in fact just takes any data received over UDP and retransmits using TCP, though it was originally written with TUIO in mind.

- It is our understanding that Flash TUIO implementation has supported UDP since 2010. So it's already possible "to use Flash directly with all standard TUIO/UDP implementations," assuming you're using a recent enough version of Flash. The TUIO specification mentions that for the next revision the reference implementation they provide will support TUIO/FLC and TUIO/TCP, though, so it would seem there's still value to supporting the other methods, perhaps for older Flash apps.

Adobe Air
A customer reported difficulties in getting Adobe Air applications to work with flash and sent a couple of non working example programs.

We did manage to get TUIO working in Flash. The problem was twofold, either trying to receive TUIO touches using UDP, the support of which is still questionable although Adobe AIR it is allegedly supported, or trying to use Flash's LocalConnection interface, which from our research appears to be very troublesome.

What worked was to use the UDP to TCP bridge documented above and then program the Flash app to receive touches through TCP.

We needed to modify a supplied ActionScript (.as file) to connect to port 3000 because, by default, udp-tcp-bridge routes udp packets from port 3333 to port 3000. This can be changed, though.

It should be noted that our initial investigations got this working in Flash's debug environment. It did not work once publishing the flash movie to a standalone swf file. Apparently Flash doesn't allow swf files located in the local file system to open connections to sockets. We didn't investigate this issue further but it looks like there's ample documentation available on the web on how to get around this issues.

We also found that one of the supplied Air application test example was able to receive pinch gestures from both Apple trackpads and from UPDD Gestures, so there's also that option for Air applications. Again, we don't think it works when running as a Flash movie but we can investigate further if required.

Contact

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