Diebold Nixdorf

User Manual

UTD

Universal Touch Driver

Linux (WNLPOS)

UTD

Universal Touch Driver

Linux (WNLPOS)

User Manual

Edition 2.3.1 Revision 1, March 2023

Content

Introduction

The Universal Touch Driver for WNLPOS 3 and WNLPOS 4 supports the Diebold Nixdorf touch screens D1xxx /pcTouch, D2xxx /pcTouch, BA9x /pcTouch, BA9x /rTouch and the touch controllers of Diebold Nixdorf All-In-One systems BEETLE / iPOS plus (Advanced/XL/SL) , BEETLE A1xxx, BEETLE /iScan, Diebold Nixdorf kiosk and others (see chapter “Identification of Touch Monitors” on page 6).

It includes the following features:

- Different touch modes
- Beep and sound on touch
- Multi monitor support
- Calibration
- Edge acceleration
- Touch delay
- Left click duration

This manual applies to version 2.3.1 of the driver.

Used Symbols

Notes and important information in this manual are marked by this symbol.

Identification of the Driver Version

Run the command `rpm -qi wntouch` identify the installed driver version.

Identification of Touch Monitors

This driver supports the following Diebold Nixdorf touch screens. They can be identified by the name, the type label or by its USB IDs. The USB ID can be read in the operating system.

Touch screen name	Type label	USB IDs
- D2150 - D2156 - D1185, A12xx - D1150, A10xx - D1156, A11xx		Vendor=222A Product=546A Product=546B Product=546C Product=546D Product=546E
- D1101 - D1150, A10xx - D1156, A11xx		Vendor=0EEF Product=C11C or Product=C11E or Product=C11D
- BA90 /pcTouch - BA92 /pcTouch - BA93w /pcTouch - iPOS+ /pcTouch		Vendor=0eef Product=7200 Product=B10D Product=B114 Product=B115 Product=B116 Product=C116 Product=C113 Product=C114
- BA92 /rTouch - BA93 /rTouch - BA9x /rTouch		Vendor=0eef Product=0001

BA91w /pcTouch		Vendor=0eef Product=B10C or Product=B110
- BEETLE iPOS+ Advanced /rTouch - BEETLE iPOS+ Advanced /pcTouch - BEETLE iPOS+ Advanced /pcTouch 10pt		Vendor=0eef Product=0001 or Product=7201 or Product=B100 or Product=B10E or Product=B10F
- BEETLE iPOS+ SL - BEETLE iPOS+ XL		Vendor=0EEF Product=B113 or Product=C111
- Kiosk Display	15" Display 19" PCAP Display 22" PCAP Display 27" PCAP Display 32" PCAP Display	Vendor=0EEF Product=72C4 or Product=7904 or Product=A100 or Product=C000
- BA82 /irTouch - BA82 /e /irTouch - BA83 /irTouch - BA83 /e /irTouch -		Vendor=6615 Product=0012
- BA80 /rTouch -		Vendor=1bfd Product=1688

This version of the **UTD does not support** the Touch Monitors of the BA82 /cTouch, BA83 /cTouch and BA7x family.

For **WNLPOS 3** some displays (e.g. Kiosk Display) require the **latest kernel version** to support these displays.

Installation of the Touch Driver on WNLPOS3 and WNLPOS4

- On WNLPOS 3: Use files from `centos65-32\noarch` and `centos65-32\i686`
- On WNLPOS 4: Use files from `centos7\noarch` and `centos7\x86_64`

Installing RPM

- open a terminal / shell as root users
- delete any preinstalled wntouch packages
  >> `rpm -e wntouch`
  >> `rpm -e wntouch-support`
  >> `rpm -e wntouch-udev`
  or whatever name the installed rpm has when listed by
  >> `rpm -qa | grep touch`
- install platform-independent scripts and settings (e.g. `wntouch-udev-xyz.rpm`)
  >> `rpm -i wntouch-udev-xyz.rpm`
  (or: `rpm -iv wntouch-udev-xyz.rpm`)
  or whatever name and version the rpm has
- install current touch driver and support package (e.g. `wntouch-xyz.rpm` and `wntouch-support-xyz.rpm`)
  >> `rpm -i wntouch-xyz.rpm wntouch-support-xyz.rpm`
  (or: `rpm -iv wntouch-xyz.rpm`)
  or whatever name and version the rpm has
- The wntouch daemon process will be started automatically by the rpm installation process. To (re-)start the daemon manually run
  >> `/etc/rc.d/init.d/wntouch restart`
  The resulting output should contain '[OK]'
  Please note that some settings on older OS versions require the X-Server to be restarted just once after the initial installation, in order for the device auto detection to become effective.

There are also instructions for “Updating from wntouch 2.0” in the “Troubleshooting” chapter.

Configuring the Daemon

The directory `/etc/opt/wn/wntouch` will be filled with template files during installation. If it is empty, the driver will still be able to run, but using default settings (no multi-screen setup, no advanced settings, no calibration).

The easiest way to change the initial settings is opening a terminal and running the touchscreen configuration tool as administrator

>> `sudo /opt/wn/wntouch/bin/wnconfig`

Select the respective devices you want to use, touch mode, beep mode and the like and click 'OK'.

Note how the two files `/etc/opt/wn/wntouch/devices` and `/etc/opt/wn/wntouch/options` have been created.

Configuring the Touch-Input device and Monitor mappings

The supporting tool for mapping input devices on monitors, thus scaling the devices resolutions with the monitor screen offset and its own resolution for a perfect match between screens touch position and graphical cursor, is `/opt/wn/wntouch/bin/wnmapping`.
This tool is also called by the wntouch daemon and the systems hardware detection daemon udev if hardware changes occur. An initial setting can be created manually by

>> sudo `/opt/wn/wntouch/bin/wnmapping --list --out`

which appends a suggestion for the currently detected setup to the mappings configuration file `/etc/opt/wn/wntouch/mappings` , which should then be edited to reflect the real situation.

Run `/opt/wn/wntouch/bin/wnmapping` again after changing file `/etc/opt/wn/wntouch/mappings`.

In some graphic configurations it can be necessary to call additional xinput commands. For these cases a script-hook to `/etc/opt/wn/wntouch/postmapping.sh` is implemented in `wnmapping`. A sample can be found in the same directory `/etc/opt/wn/wntouch/postmapping.sh.template`

Device Configuration File

The device file is needed to define a specific order in which the driver addresses the touch controllers. This is important to configure multi-screen setups. The file is located at `/etc/opt/wn/wntouch/devices`. It can be created by the configuration tool (see above).

It has to contain the sysfs_paths of the devices to be handled by the driver daemon, which represent the USB ports the touch devices are connected to. These are handed to the daemon as command line parameters by the daemon start script.

The device file should be empty in case only one touch screen is connected to the system.
The devices configuration file needs to be created again if there is any change in the USB port configuration.

If you intend to create it manually, you can run (as root user) the following command to get a list of supported devices in the devices file, and then deleted the devices you don't need or want.

```
>> /opt/wn/wntouch/sbin/wntouch --list >
/etc/opt/wn/wntouch/devices
```

`/etc/opt/wn/wntouch/devices` example:

```
/sys/devices/pci0000:00/0000:00:1a.7/usb1/1-2/1-2.2/1-
2.2.1/1-2.2.1:1.0
/sys/devices/pci0000:00/0000:00:1a.7/usb1/1-2/1-2.1/1-
2.1.1/1-2.1.1:1.0
```

Options File

The options file is located at `/etc/opt/wn/wntouch/options`.
It can be created by the configuration tool (see above).
It contains options handed to the driver as command line parameters by the daemon start script.
It may contain the following options and settings:

Setting the touch mode

`TOUCH_MODE=(touch|release|mouse|multitouch|passthrough)`
The default mode is 'release'

`touch` Execute click event when finger touches the screen.

`release` Execute click event when finger is released from the screen.

`mouse` Execute pressed, moved and released events like a mouse. In this mode, the "right mouse button" emulation can be activated.

`multitouch` Pass multitouch events (simultaneously pressed locations on screen in parallel), still applying calibration and filters such as edge acceleration

`passthrough` Pass through all events from the attached touch device unfiltered, no calibration and edge acceleration are applied.

Beep on click settings

Option a – global setting:

`BEEP_PER_DEVICE` has to be commented out with #
be not present in the options file be initialized to 0, no, off

Option a1 – PC-Speaker:
`BEEP_TYPE=system`
`BEEP_DEVICE=<console>` (e.g.: `BEEP_DEVICE= /dev/console`)

Option a2 – Soundcard:
`BEEP_TYPE=alsa`
`SND_FILE=/path/to/file/sound.wav`

PCM_NAME=<sound device> (e.g.: PCM_NAME=default or PCM_NAME=hw:1,0)

Option b – setting per touch device

Please note, that the parameters have to be numbered with _1 und _2 and not with _0 und _1.

BEEP_PER_DEVICE=(1|TRUE|ON|YES)

Option b1 – PC-Speaker:

BEEP_TYPE_1=system and/or
BEEP_TYPE_2=system

BEEP_DEVICE_1=<console> and/or
BEEP_DEVICE_2=<console>
(e.g.: BEEP_DEVICE_1= /dev/console)

Option b2 – Soundcard(s):

BEEP_TYPE_1=alsa and/or
BEEP_TYPE_2=alsa

SND_FILE_1=/path/to/file/some_sound.wav and/or
SND_FILE_2=/path/to/file/another_sound.wav

PCM_NAME_1=<sound device> and/or
PCM_NAME_2=<sound device> (e.g.: PCM_NAME_2=default or PCM_NAME_2=hw:1,0)
Combinations of b1 and b2 are possible.

Edge acceleration settings

Option 1 – global settings:

EDGE_RANGE_V=<edge height in Y direction [screen px]>
(e.g.: EDGE_RANGE_V=50)

EDGE_RANGE_H=<edge width in X direction [screen px]>
(e.g.: EDGE_RANGE_H=73)

EDGE_MUL=<acceleration multiplier for all edges>
(e.g.: EDGE_MUL=1.5)

Option 2 – settings per touch device

Please note, that the parameters have to be numbered with _1 und _2 and not with _0 und _1.
`EDGE_PER_DEVICE=(1|TRUE|ON|YES)`
Set the following values (if desired) as described above:

```
EDGE_RANGE_V_1=
EDGE_RANGE_H_1=
EDGE_MUL_1=

EDGE_RANGE_V_2=
EDGE_RANGE_H_2=
EDGE_MUL_2=
```

Device selection settings

The daemon will try to find input queues that correspond to the sysfs paths provided on the command line. If started via runcontrol script these are the ones provided in `/etc/opt/wn/wntouch/devices`.
If there are less than two devices given in, it will try to find further supportable devices connected to the system.
Please be aware that the devices given take up their respective slots even if not connected to the system, thus modifying the order and gaps in the numbering of emulated wntouch0 … wntouch7 devices.

In order to avoid devices not defined on the command line from being supported, use the following setting:

`STRICT_DEVICE_SELECTION=(1|TRUE|ON|YES)`

Touch/Release delay settings

The touch/release delay will prevent short touches (in click-on-touch mode) and releases (in click-on-release mode) to issue a click event.

The default setting for the delay is 50 msec.
The setting can be adjusted with the following parameter `TOUCH_DELAY=<touch/release delay [msec]>` (e.g. `TOUCH_DELAY=75`)
The minimum delay is 0 msec.
The maximum is 300 msec.

MinLeftclickDuration setting

In click-on-touch mode the minimum left click duration defines the time (in ms) between the faked touch and release events passed to the OS when the touch device was touched.

The default setting is 10 msec.
The setting can be adjusted with the following parameter
`MIN_LEFTCLICK_DURATION=<min. left click duration [msec]>`
`(e.g. MIN_LEFTCLICK_DURATION=15)`
The minimum value is 0 msec.
The maximum value is 2000 msec.

MaxLeftclickDuration setting

In click-on-release mode the maximum left click duration defines the time (in ms) between the faked touch and release events passed to the OS when the touch device was released.

The default setting is 200 msec.
The setting can be adjusted with the following parameter
`MAX_LEFTCLICK_DURATION=<max. left click duration [msec]>`
`(e.g. MAX_LEFTCLICK_DURATION=250)`
The minimum value is 0 msec.
The maximum value is 2000 msec.

Debug Level setting

The wntouch daemon can be set to different levels of writing runtime information to its logfile at `/var/log/wn/wntouch/wntouch.log` using the `DEBUG_LEVEL` variable:

`0` - Nothing will be logged
`1` - Fatal errors and attached devices (default)
`2` - Also show warnings
`3` - Also show events and decoded touch/mouse protocol information

Example

`/etc/opt/wn/wntouch/options` example:

```
TOUCH_MODE=touch
DEBUG_LEVEL=2
EDGE_RANGE_H=75
EDGE_RANGE_V=75
EDGE_MUL=1.5
BEEP_TYPE=system
TOUCH_DELAY=20
MIN_LEFTCLICK_DURATION=15
MAX_LEFTCLICK_DURATION=250
```

Daemon Calibration

After installation and configuration, touch devices should match the corresponding monitors. In rare cases, touch position and cursor position may be off by a few pixels. In these cases, the calibration tool can be used while the wntouch daemon is running:

```
>> /opt/wn/wntouch/bin/wncalibrate
```

By now, there should be files `/etc/opt/wn/wntouch/calibrationX` for every active device (X is the device number: 'calibration0', 'calibration1' and so on).

The calibration files contain position and size of

- the virtual screen (measured in touchscreen coordinates)
- the actual screen (measured in display pixel)
- the virtual screen (measured in display pixel)

If the calibration was properly conducted, all necessary calibration data is written to the respective calibration files.

The calibration should not be used to assign the touch controllers to certain monitors. This is done by the wnmapping tool, which will be called by the wntouch daemon and udev automatically. Failure to match a touch device with a connected

monitor is usually caused by a misconfiguration of the `mappings` or `devices` file in `/etc/opt/wn/wntouch`.

See chapter “Calibration Tool” for more information.

XServer Configuration

Now even though the driver daemon might be running (after a restart with the correct settings), the X11 server may not yet know about the inputs it provides.

During installation, a configuration file is being added to `/etc/X11/xorg.conf.d/`:

```
Section "InputClass"
        Identifier  "wntouch"
        Driver      "evdev"
        MatchDevicePath   "/dev/input/wntouch*"
EndSection
```

This entry tells the X-Server to automatically add devices emulated by the wntouch daemon. Other entries `/etc/X11/xorg.conf` in are not needed.

In CentOS 6.5, the X-Server will need help from the HAL (Hardware Abstraction Layer) daemon in order to recognize newly attached devices during runtime. For this purpose another configuration file for HAL is provided in the wntouch package, `/usr/share/hal/fdi/policy/20thirdparty/99-wntouch.fdi`:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<deviceinfo version="0.2">
  <device>
    <match key="info.category" contains="input">
      <match key="info.product"
contains_outof="wntouch">
```

```
<merge key="input.x11_driver"
type="string">evdev</merge>
<append key="info.capabilities"
type="strlist">tablet</append>
      </match>
    </match>
  </device>
</deviceinfo>
```

This file is no longer needed in CentOS 7, but still kept for reference.

If the wntouch packages were installed for the first time after the X-Server has already been started already, it may be necessary to restart X11 because the X-Server only reads its configuration files once during start up.

Touchcontroller vs. Monitor Configuration

`/opt/wn/wntouch/bin/wnmapping` is a tool for mapping touch controllers to monitors, automatically setting resolution and position. Its configuration file is `/etc/opt/wn/wntouch/mappings`, containing combinations of touch controller and monitor socket names, separated by a single space character, no quotes. The available monitor socket names depend on the graphic chip and can be listed with the `xrandr` command. A sample initial setup based on “first detected, first listed” can be created by

```
>> sudo /opt/wn/wntouch/bin/wnmapping --list --out
```

which will append the current findings to `/etc/opt/wn/wntouch/mappings`.

The wnmapping tool has several options which will be used in different scenarios:

```
>> /opt/wn/wntouch/bin/wnmapping --help
Usage: /opt/wn/wntouch/bin/wnmapping options
 -a   | --auto        Automatic choice of wntouch-only or general mappings
 -l   | --list        Print autoconfigured mappings
 -o   | --out         Append current mappings to config file
 -n   | --nolvds      Skip internal LVDS entries
```

```
-r   | --reverse     Reverse order of screens
-w   | --wntouch     Only map wntouch devices from mappings file
-d   | --debug       Print verbose debug info
-t s | --time s      Wait s seconds
--disable          Detach non-wntouch devices from X server
--enable           Re-Attach non-wntouch devices to X server
-h | --help        This help
```

If called manually with no parameters, wnmapping will map all touchdevice - monitor combinations in `/etc/opt/wn/wntouch/mappings` and add a “best guess” mapping for those devices not listed in the config file. It can also be used independently of the wntouch daemon in order to map the “raw” touch controller devices.

It is not harmful to list both wntouchX as well as the original device names in /etc/opt/wn/wntouch/mappings, as raw devices attached to wntouch-provided devices will be automatically get deactivated by wntouch as long as the daemon is running.

A sample `/etc/opt/wn/wntouch/mappings` file:

```
eGalax Inc. eGalaxTouch EXC7910-1265-53.00.00 HDMI-0
wntouch0 HDMI-0
```

With version 2.3 the search algorithm for the touch controllers has been changed to improve the detection speed significantly. The order in which the touch controllers are found (and the device name wntouchX) can be changed by that if there was no `devices` file created.
Please check the device names used in the `mappings` file if the package is updated to version 2.3 or later.

Right-Click Emulation

`/opt/wn/wntouch/bin/wnmapping` calls `/opt/wn/wntouch/bin/wnemulate3` in order to emulate a “menu button click” using a special Xorg xinput2 layer feature. As of the current state of Xorg, this only works in “mouse” mode, since multi touch events are just

passed through to the application layer, and the application has to decide by itself on emulating a different button on different gestures, which the gnome desktop is known to do when long-pressing on the desktop background in order to display a menu.

In wntouch “mouse” mode, xinput2 will modify touch events to become mouse click events in order to get application-independent “right clicks” using the “emulate menu” feature.

The supplied sample `/etc/opt/wn/wntouch/rightclick` file explains the meaning of the given variables:

```
# Emulate menu button by long click? (1=on, 0=off)
# This only works in "mouse" mode
# EMULATE_MENU=0
# How long to press for detecting right mousebutton (ms)
# EMULATE_MENU_TIMEOUT=3000
# Jitter elimination (virtual pixels per duration)
# EMULATE_MENU_THRESHOLD=1000
```

Restart

Shutdown the X11 server, restart the wntouch daemon and restart the X11 server. This could also be done by a system reset (e.g. `init 6`).

Quicker would be

- to log out the current user
  (end the X11 server and fall back to console)
- stop the wntouch daemon

```
>> /etc/rc.d/init.d/wntouch stop
```

- start the wntouch daemon

```
>> /etc/rc.d/init.d/wntouch start
```

- restart the X11 server

```
>> startx
```

or

```
>> init 5
```

Note that restarting X is only necessary the first time after the `/etc/X11/xorg.conf.d/99-wntouch.conf` config file has been installed. For other changes in the configuration of wnconfig, it is sufficient to just restart wntouch by its start/stop script:

```
>> sudo /etc/rc.d/init.d/wntouch restart
```

which will be done automatically when saving a new configuration using the wnconfig tool.

Troubleshooting

Updating from wntouch 2.0

- Entries in `/etc/X11/xorg.conf` are no longer needed. Remove all entries where "Device" is "/dev/input/wntouchX".

```
InputDevice "Wntouch0" "SendCoreEvents"
InputDevice "Wntouch1" "SendCoreEvents"
```

  The identifier might also be different (e.g. `Wntouch_egalax0`). See chapter “XServer Configuration” for the current configuration method which does not need manual configuration.

- The assignment of touch controller and monitor is not done by calibration anymore. This has

the advantage that it is independent from the resolution and position of the displays. Rename or remove
`/etc/opt/wn/wntouch/calibration*`
See chapter “Configuring the Touch-Input device and Monitor mappings” and “Daemon Calibration” for more information

Pointers and screens don’t match

If after restart the pointer moves out on the other screen,
you probably only have to exchange or remove the `calibrationX` files in folder `/etc/opt/wn/wntouch`.

or re-run the calibration.

Beep-on-Click Wav-File mixing

If the beep-on-click functionality of the driver shall be used, it might be necessary to create PCM sound devices that point to the desired PCM hardware, but feature mixing capabilities. Otherwise no click sound can be played until the last has been fully completed.

After installation the file `/etc/asound.conf` includes the file `/etc/alsa/pulse- default.conf` that looks like:

```
# PulseAudio plugin configuration
# $Id: pulse-default.conf,v 1.3 2008/03/09 15:50:49
lkundrak Exp $

# Let's create a virtual device "pulse" for mixer and
PCM

pcm.pulse {
       type pulse
       hint {
              description "PulseAudio Sound Server"
       }
}

ctl.pulse {
       type pulse
       hint {
```

```
                description "PulseAudio Sound Server"
        }
}

# Let's make it the default!

pcm.!default {
        type pulse
        hint {
                description "Default"
        }
}


ctl.!default {
        type pulse
        hint {
                description "Default"
        }
}
```

and some lines should be added to make it look like:

```
# PulseAudio plugin configuration
# $Id: pulse-default.conf,v 1.3 2008/03/09 15:50:49
lkundrak Exp $

# Let's create a virtual device "pulse" for mixer and
PCM

pcm.pulse {
        type pulse
        hint {
                description "PulseAudio Sound Server"
        }
}

ctl.pulse {
        type pulse
        hint {
                description "PulseAudio Sound Server"
        }
}

ctl.!default {
        type pulse
                                                hint {
```

```
                description "Default"
        }
}

pcm.dmixer0 {
        type plug
        slave.pcm "dmix:0,0"
}

pcm.dmixer1 {
        type plug
        slave.pcm "dmix:1,0"
}

pcm.!default {
        type plug
        slave.pcm "dmix:1,0"
        hint {
                description "Default"
        }
}
```

Now the two devices '`dmixer0`' and '`dmixer1`' can be used for output to the two available sound cards (internal and USB).
Please note that you might need to adapt this to your system setup and soundcards available and/or desired for output.

The Touch Driver in Detail

Architecture:

The touch controllers are exposed as evdev devices in `/dev/input/eventX`. By default they are recognized as mouse by the kernel and all events are routed to a `/dev/input/mouseX` device which is also automatically used by the xorg server. Some newer touchscreen devices may not be recognized at all by the X-Server though, because of an unknown product/vendor id, these will only be functional with a version of wntouch that knows and translates their events to mouse clicks or touch events.

The raw input events generated by the original device files do not allow any
calibration and advanced processing support, therefore wntouch inserts an additional abstraction
layer by means of the developed driver. As stated by the requirement, this abstraction layer is completely in user-space and implemented with a daemon program called **wntouch**. wntouch will scan all devices in `/dev/input/` for supported touchscreen controllers. Whenever it finds a matching input device it will open it and request for exclusive access. This will stop further processing of the incoming events by the kernel and conversion to mouse events. To allow Xorg to use the device, wntouch will create a new evdev device named `wntouchX` (the internal evdev name of a device can be seen using the evtest tool). Additionally it will create a link in `/dev/input/wntouchX` to the newly created `eventX` device so that it can be referred directly in configuration files.

With version 2.1.x of the driver, an accompanying user space tool called **wnmapping** is used to map monitors and touch devices correctly, using the xinput2 component of Xorg, so the former manual screen calibration becomes optional. Also, second mouse button clicks can be emulated in mode “mouse” by long presses on the touchscreen surface, which is configured in `/etc/opt/wn/wntouch/rightclick` (default off).

The wntouch daemon is controlled by command line options which are parsed from variables in `/etc/opt/wn/wntouch/options`, which are

evaluated and assembled into one large commandline by `/etc/init.d/wntouch start` on system boot time.

Mapping Tool

In order to allow automatic configuration changes and mappings of newly attached devices, the wnmapping tool is called by udev each time a new supported device is detected. Additionally, the wntouch daemon gets to re-enumerate its devices by receiving signal 1 (SIGHUP) from wnmapping, when called by udev. In the reverse scenario, whenever wntouch needs to refresh its emulated device list, wnmapping will be called with options `--wntouch` and `--disable`, so only the wntouch devices are attached to screens by Xorg, and the original "raw" devices are detached from the input queue in order to avoid conflicting mouse/touch protocol data.

Calibration Tool

In order to allow calibration, the wntouch driver exposes a calibration interface through a socket for each device discovered located in `/var/run/wn/wntouch/wncalibX`.
Access rights to the calibration interface can be controlled by setting the desired access rights to these special files.

The calibration tool wncalibrate will use these sockets to communicate with the driver and set new calibration data. Once opened the wntouch driver will send all incoming events to the calibration socket unfiltered instead of sending them to the output event device. wncalibrate will use the data to calculate a new calibration and will send this new data to the wntouch driver using the following commands:

```
> CALIBRATE <min_x> <max_x> <min_y> <max_y>
> SCREEN_DATA <x> <width> <y> <height>
> DESKTOP_DATA <x> <width> < y> <height>
```

Once new calibration data is sent, the wntouch driver will generate output events again using this calibration data. In order to persist the new calibration data the wncalibrate tool needs to send the following command:

```
> SAVE
```

If the connection to the wntouch driver is disconnected before sending the save command, the original calibration is restored by the wntouch driver. The wntouch driver will store the calibration data as text file in

`/etc/opt/wn/wntouch/calibrationX`, containing the min_x, max_x, min_y and max_y values on individual lines.

The calibration tool can be started as follows:

```
>> /opt/wn/wntouch/bin/wncalibrate --screen=0
>> /opt/wn/wntouch/bin/wncalibrate --screen=1
```

The calibration tool has a draw mode to verify the calibration. This can be started with the following Parameter:

```
>> /opt/wn/wntouch/bin/wncalibrate --test
```

This mode can also be used with the `--screen` parameter.

Additional Configuration Files

As each login display manager uses its own startup script before the user logs in, and also a users profile after login may change screen geometry and resolution, additional scripts take care of calling wnmapping again after such events. They are automatically installed by the wntouch-udev package.

`/etc/gdm/Init/Default /etc/X11/xdm/Xsetup /etc/kde4/kdm/Xsetup /etc/kde5/kdm/Xsetup`: Configuration files for display managers gdm, xdm, kde4 or kde5

`/etc/X11/xinit/xinitrc.d/zz-wntouch.sh`: Executed during session startup after login.

`/etc/xdg/autostart/zz-wntouch.desktop`: Executed after other user-defined autostartup-scripts.

If **no window manager** if used, the startup script might not be called. In this case, the mapping is not applied. Add a call of `/etc/X11/xinit/xinitrc.d/zz-wntouch.sh` to the default start script e.g. `~/.xinitrc`

Driver Version History

- 2.1.8 – Initial release of WN-UTD 2.1
- 2.1.9 – Added hook-script support for mapping
- 2.1.10 – Fixed race condition resulting in high CPU load
- 2.2.2 – Support for iPOS+ SL, XL, BEETLE A1xxx and D1xxx added
- 2.2.3 – Support for Kiosk Display
- 2.3.1 – Support D2xxx Displays. Device detection improved (See Touchcontroller vs. Monitor Configuration)

Diebold Nixdorf Systems GmbH
D-33094 Paderborn