

# Software manual

For Elitac Tactile Displays  
Part 2 of 3

***This manual describes the software to be used with Elitac Tactile Displays. It is part of a suit of three manuals:***

***Part 1 – it describes the general use of Elitac Tactile Displays***

***Part 2 – this manual***

***Part 3 – it describes the specific use of the Elitac Tactile Display you acquired***

Manual Version:	3.3
HIDcom version:	HIDcom_V2.0 and newer
Date	December 2017
Technical inquiries	Bram Bicknese support@elitac.nl
Business inquiries	Merijn Klarenbeek m.klarenbeek@elitac.nl

Elitac B.V.  
info@elitac.nl  
Amsterdamsestraatweg 285  
3551CE Utrecht  
The Netherlands

KvK nr. 56506597  
BTW nr. NL 852160033B01

Copyright © 2017 Elitac B.V. All rights reserved. No part of this document may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior written permission of Elitac B.V.

## Table of Contents

1. How does the Tactile Display work? .....	4
2. Connect the Tactile Display to a computer .....	5
2.1. Install Elitac HIDCom.....	5
2.1.1. Windows (Windows 7, Windows 10).....	5
2.1.2. Mac OS X .....	5
2.1.3. Linux.....	6
3. Controlling the Tactile Display from your own application.....	8
4. How to create tactile patterns? .....	11
4.1. Create a Tactile Pattern.....	11
4.2. How to Format pattern files .....	12
4.3. Priority of tactile patterns between <code>ArrayOfAction</code> .....	13
5. Body Surface Coordinate System .....	14
6. Tactor Constellations and Tactile Displays .....	16
7. Troubleshooting .....	17
7.1. Elitac HIDCom Software.....	17
7.2. Patterns .....	17
8. Technical support .....	17
9. Register .....	18
10. Glossary .....	18

## 1. How does the Tactile Display work?

The Elitac Tactile Display uses the skin to display information using localised vibration. It consists of one or more small vibration elements, called tactors, that are mounted in textile and pressed to the skin. By activating the tactors in spatiotemporal patterns, called tactile patterns, information can be displayed. The Elitac tactile hardware consists of a control module with up to 32 tactors. They can be arranged in many shapes or spatial constellations inside textiles, called tactor constellations.

Tactors can be controlled individual using the corresponding tactor number. Or a set of tactors can form together a tactile display that can be used to display information on a specific location on the skin Body Surface Coordinate System . The location on the skin can be described by our proprietary body coordinate system.

The Tactile Display can be controlled through a wired USB or wireless Bluetooth connection with software running on a host device (PC, smartphone, tablet) that runs on either one of the following operating systems: Windows; Mac OS X; or Linux. The software consists of two components: *Elitac HIDCom* that controls the hardware directly and listens to a UDP port; and your or third-party software (e.g. *Elitac\_UDPClient\_Example.jar*) that sends commands to this UDP port. All this combined makes the Elitac Tactile Display an easily customizable display that can be adapted for a wide range of applications.

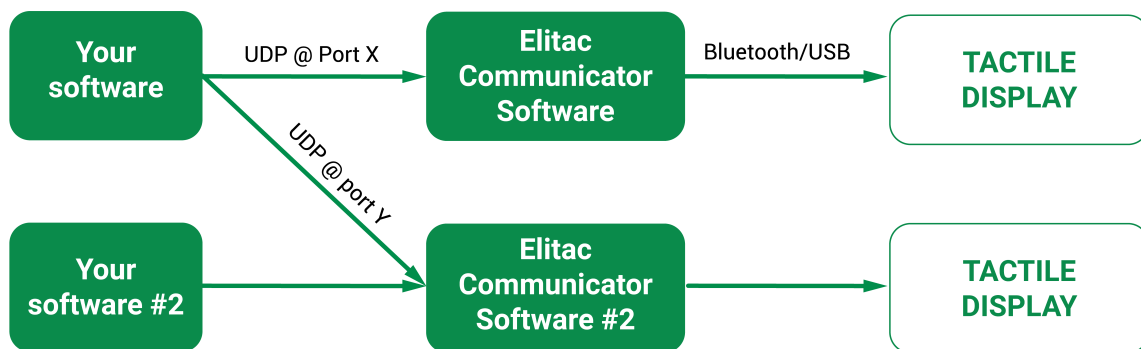


Figure 1 Schematic depicting the control of the Elitac Tactile Display: Your software or third-party software sends commands to the Elitac HIDCom software through UDP, which is listening to an adjustable port number. The Elitac HIDCom sends these commands to the control module through either a Bluetooth® or USB connection. Your software can talk simultaneously to more than one copy of Elitac HIDCom through different UDP ports. When these copies run on a single host device Bluetooth ® 4.0 or USB connections are required.

## 2. Connect the Tactile Display to a computer

The control software of the Elitac display, called *Elitac HIDCom*, is compatible with Windows 7, Windows 10, Mac OS X, Linux. It allows you to play predefined patterns.

The *Elitac HIDCom* is a piece of software written in Java that allows the user to play predefined tactile patterns that are stored in XML files. The software can be controlled by sending messages through UDP. This allows it to run on many platforms and it can be controlled from many applications written in e.g. Java, C#, C++, Python, Matlab, or Simulink.

- ! When connecting the Tactile Display to your own application, ensure that your application and *Elitac HIDCom* are running on the same operating system. UDP messages can be send to localhost (IP-address 127.0.0.1).
- ! *Elitac HIDCom* can be controlled from a remote computer, but this will add a network dependent delay.

### 2.1. Install *Elitac HIDCom*

Follow the instructions below install *Elitac HIDCom* on your operating system.

Note: Images (on next page) may appear differently according to the version and personal settings of your operating system.

#### 2.1.1. Windows (Windows 7, Windows 10)

1. Check if Java Runtime Environment (JRE) 7 or 8 is installed. If not, download it from e.g. [Oracle](#), and install it.
2. Copy the control software named *Elitac\_HIDCom.jar* to a directory of your choice.
3. Double click on *Elitac\_HIDCom.jar* to run it.
  - a. A pop-up from Windows Firewall will show; give the software permission to communicate through private and public networks.
  - b. The *Elitac HIDCom* icon will appear in the system tray. The software is running.
  - c. The connection status is shown when hovering over it. Figure 2
4. Click on the Elitac HIDCom icon and right-click on "Settings" Figure 3
  - a. UDP Settings: Modify the UDP settings according to Figure 4
  - b. Pattern folder:
    - i. Click "Open"
    - ii. Go to the folder that contains tactile pattern files.
    - iii. Select the folder and click "Choose" (you can only choose folders).
  - c. **USB:** Plug the USB-cable into the device and computer (if not already), a connection is automatically made.
  - d. **Bluetooth:** Insert the BLE112 Bluetooth dongle. It will be automatically installed.
  - e. **Bluetooth:** Check in "Devices and Printers" which COM port number the dongle was given on Windows. (Right click on the device, select "properties" and check the "hardware" tab.)
  - f. **Bluetooth:** Enter the COM port number in the "BLE112 settings" (Figure 4) using "COM<#>".
  - g. **Bluetooth:** Click the "BLE112 Connect" button. The dongle will now discover Elitac Bluetooth enabled devices. A list of compatible devices is shown. Select the device with the serial number that is printed on the label of your Elitac hardware.

#### 2.1.2. Mac OS X

1. Check if Java Runtime Environment (JRE) 7 or 8 is installed. If not, download it from e.g. [Oracle](#), and install it.
2. Copy the control software named *Elitac\_HIDCom.jar* to a directory of your choice.
3. Double click on *Elitac\_HIDCom.jar* to run it.
  - a. The *Elitac HIDCom* icon will appear in the system tray. The software is running.
  - b. The connection status is shown when hovering over it. Figure 2
  - c.
4. Click on the Elitac HIDCom icon and click on "Settings" Figure 3

- a. UDP Settings: Modify the UDP settings according to Figure 4
- b. Pattern folder:
  - i. Click "Open"
  - ii. Go to the folder that contains tactile pattern files.
  - iii. Select the folder and click "Choose" (you can only choose folders).
- c. **USB:** Plug the USB-cable into the device and computer (if not already). Switch the device on. A connection is automatically made.
- d. **Bluetooth:** Insert the BLE112 Bluetooth dongle. It will be automatically installed.
- e. **Bluetooth:** use "lshw -short" command in a terminal to find the device name of the dongle on MacOS.
- f. **Bluetooth:** Enter the device name in the "BLE112 settings" (Figure 4).
- g. **Bluetooth:** Click the "BLE112 Connect" button. The dongle will now discover Elitac Bluetooth enabled devices. A list of compatible devices is shown. Select the device with the serial number that is printed on the label of your Elitac hardware.

### 2.1.3. Linux

1. Check if Java Runtime Environment (JRE) 7 or 8 is installed. If not, download it from e.g. [Oracle](https://www.oracle.com/in/java/technologies/javase-downloads.html) and install it.
2. Copy the control software named *Elitac\_HIDCom.jar* to a directory of your choice.
3. Double click on *Elitac\_HIDCom.jar* to run it.
  - a. The *Elitac HIDCom* icon will appear in the system tray. The software is running.
  - b. The connection status is shown when hovering over it. Figure 2
4. Click on the Elitac HIDCom icon and click on "Settings Figure 3
  - a. UDP Settings: Modify the UDP settings according to Figure 4
  - b. Pattern folder:
    - i. Click "Open"
    - ii. Go to the folder that contains tactile pattern files.
    - iii. Select the folder and click "Choose" (you can only choose folders).
  - c. **USB:** Plug the USB-cable into the device and computer (if not already), a connection is automatically made.
  - d. **Bluetooth:** Insert the BLE112 Bluetooth dongle. It will be automatically installed.
  - e. **Bluetooth:** use "sudo lshw -short" command in a terminal to find the device name of the dongle on Linux.
  - f. **Bluetooth:** Enter the device name in the "BLE112 settings" (Figure 4).
  - g. **Bluetooth:** Click the "BLE112 Connect" button. The dongle will now discover Elitac Bluetooth enabled devices. A list of compatible devices is shown. Select the device with the serial number that is printed on the label of your Elitac hardware.



Figure 2 - *Elitac HIDCom* Icon in system tray showing connection status

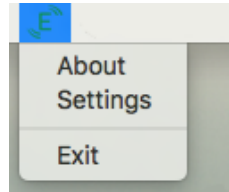


Figure 3 – *Elitac HIDCom* menu

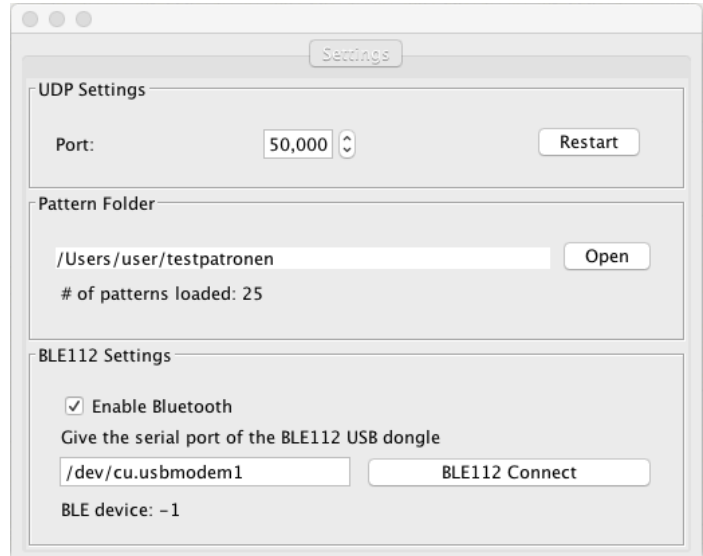


Figure 4 – *Elitac HIDCom* Settings for Windows, Mac OS X and Linux.

#### Modifying *Elitac HIDCom* settings:

- UDP settings allow you to set the UDP port the Elitac HIDCom listens to. Press restart after changing the port number.
- The Pattern Folder allows you to select the folder that contains the patterns.
- The Serial Settings allow you to select the tactile hardware to connect to through either a Bluetooth® or USB connection.

### 3. Controlling the Tactile Display from your own application

Before trying to control the Tactile Display from an application, ensure that the *Elitac HIDCom* Software is installed and that it has established a connection with the Tactile Display ([2.1 Install Elitac HIDCom](#)).

#### **Sending messages from your application to Elitac HIDCom**

There are two types of messages that you can send from your application to *Elitac HIDCom*. The first type is a request for information and it requires you to handle the answer. The general format of a request is:

```
?<COMMAND>, <ARGUMENT_1>, ..., <ARGUMENT_N>
```

The arguments are comma delimited, without spaces. *Elitac HIDCom* will always respond using the following general format:

```
<COMMAND>, <RETURNVALUE_1>, ..., <RETURNVALUE_N>
```

The second type is a command. The general format of the command is:

```
!<COMMAND>, <ARGUMENT_1>, ..., <ARGUMENT_N>
```

*Elitac HIDCom* responds to a command in the same way as a request. Depending on the command, *Elitac HIDCom* may send a return value, however you are not required to parse it.

#### **The following messages are recognized by *Elitac HIDCom*:**

```
?IsConnected
// Checks if the tactile display is connected.
// Returns <name> of connection if so, else the string "Not connected".

!PlayPattern,patternName,<optional argument1>, ..., <optional argumentN>
// Plays the tactile pattern
// "patternName" is the filename of the pattern to be played, without extension and path. The name
// should match one of the pattern names that is returned by the request ?GetPatterns.
// Returns the total duration of the pattern in ms if the pattern has been and the optional arguments
// are correct, else -1, followed by an error message string.
// Optional arguments can be used to modify a predefined tactile pattern at run-time. They are
// executed in the order listed:
    invertHorizontal=<true/false>
    // inverts the pattern around the zero circumference coordinate.

    invertVertical=<true/false>
    // inverts the pattern around the zero vertical coordinate.

    circumferenceCoorOffset=<signed integer>
    // shifts the pattern in the circumference coordinate. Range [-1023..1023].

    verticalCoorOffset=<signed integer>
    // shifts the pattern in the vertical coordinate.

    displayID=<unsigned integer>
    // displays the pattern on a different display id.

    bodyPart=[UNSPECIFIED; TORSO; LEFTUPPERARM; RIGHTUPPERARM;
    LEFTLOWERARM; RIGHTLOWERARM; LEFTHAND; RIGHTHAND; LEFTUPPERLEG;
    RIGHTUPPERLEG; LEFTLOWERLEG; RIGHTLOWERLEG; LEFTFOOT; RIGHTFOOT;
    HEAD]
    // displays the pattern on a different body part.

    horizontalAngleOffset=<float>
    // shifts the pattern in the circumference coordinate over the given angle in degrees.
```



```

compassAngle=<float>
// plays the pattern from the external compass direction in degrees. It modifies the
circumference coordinates of the pattern. If compassAngle is given, latitude/longitude is
not allowed. Requires the optional compass module.

longitude=<double>
// plays the pattern from the external geolocation given by the longitude and latitude. It
modifies the circumference coordinates of the pattern. When longitude is given, latitude has
to be given too. If longitude is given, compass angle is not allowed. Requires the optional
compass and GNSS module.

latitude=<double>
// plays the pattern from the external geolocation given by the longitude and latitude. It
modifies the circumference coordinates of the pattern. When latitude is given, longitude has
to be given too. If latitude is given, compass angle is not allowed. Requires the optional
compass and GNSS module.

intensityIncrease=<signed integer>
// changes the overall intensity of the pattern. Range [-15..15]

```

?GetPatterns

```

// request the available tactile patterns
// Returns a list of pattern names.

```

!ActivateTactor, tactorNumber, offset, duration, intensity

```

// activates a single tactor
// "tactorNumber" is the number of the tactor that should be activated. Range 1-32.
// "offset" is the offset in milliseconds from the moment the command has been received by the
device. Range 0-60000 ms; resolution 5. When it is zero, the tactor is activated immediately.
// "duration" is the duration of the activation in milliseconds. Range 0-60000 ms; resolution 5. Zero
indicates that the tactor should be switched off.
// "intensity" is a number indicating the logarithmic stimulation intensity (0-15). Zero is the lowest
setting and does not mean off. 15 means full power.
// Returns the total duration in milliseconds, defined as "offset" + "duration". When the call is
unsuccessful, it will return -1.

```

?GetBatteryStatus

```

// Gets the current battery capacitance
// Returns the current capacity in percentages. When the call is not successful it will return -1.

```

?GetFirmwareVersion

```

// Gets the firmware version
// Returns a character string containing the version number

```

?GetServerVersion

```

// Gets the software version of Elitac HIDCom.
// Returns a character string containing the version number

```

!ChangeTactorState, tactorNumber, intensity, duration

```

// Changes the state of the indicated tactor. The change will take effect at the next call of
!ChangeTactorState can be used to change tactor states in a timer callback routine. It overwrites
any previously changed tactor state for the indicated tactor, since the last !ExecuteTactorStates
call.
// "tactorNumber" is the number of the tactor that should be activated. Range 1-32.
// "intensity" is a number indicating the logarithmic stimulation intensity. Range 0-15. Zero is the
lowest ON setting and doesn't mean off. An intensity of 15 means full power.
// "duration" is the duration of the activation in milliseconds. Range 0-60000 ms; resolution 5 ms.
Zero indicates that the tactor should be switched off. We suggest a minimum duration of 100 ms.
The maximum update frequency of the states is determined by the delay of the connection to the
hardware and the minimum duration can be adjusted accordingly. To resolve any jitter in the
connection delay, we suggest to add an additional 20 millisecond to the duration (e.g. If the update
rate is 1/100 ms-1, use a duration of at least 120 ms).

```

```
// Returns TRUE or FALSE. False is returned if at least one of the arguments is invalid.

!ExecuteTactorStates
// Executes the list of tactor state changes that have been added with calls to !ChangeTactorState.
The state changes will take effect simultaneously. The list will be cleared after execution.
// Returns the number of executed states. (zero if there are no state changes in the list)

!ClearTactorStates
// Clears all currently loaded tactor states in the state list.
// Returns the number of cleared states (zero if there are no state changes in the list)
```

The basic tactile feedback implementation will monitor a sensor value at a fixed rate. Depending on the sensor value a tactile pattern will be chosen from the available patterns and sent through UDP to *Elitac HIDCom* with the `!PlayPattern` command. The sensor value could also be used to modify a predefined, template pattern using the optional arguments. Multiple tactile patterns can be played simultaneously. You have to resolve conflicts between patterns yourself. For a code example written in Java see the application *Elitac\_UDPCClient\_Example.jar*.

- ! The firewall may block the selected UDP port. Make sure the firewall allows UDP traffic over that port.

Instead of playing a predefined pattern given a sensor value, you could also determine on-the-fly which tactor(s) should be activated. The state of each tactor can be changed using the `!ChangeTactorState` command. The state changes will take effect simultaneously after calling the `!ExecuteTactorStates` command. You can use these commands to build a stateless or stateful, on-the-fly control of the tactors. Stateless control means that at each update you have to define for each tactor (that should be activated) at what intensity it should vibrate. You should use an activation duration that is slightly larger than the update period and equal for all tactors. Stateful control means that at each update you only have to define the intensity for the tactors that need to change. You should use an activation duration that is marginally larger than the update period or even indefinitely, unless you know beforehand for how long a given tactor should be activated.

## 4. How to create tactile patterns?

- ! When the formatting of a pattern file is incorrect it will not be loaded.
- ! Ensure that valid values for the variables in a pattern are used or the pattern may not play correctly.
- ! Ensure that valid tactor numbers are used in a pattern, or the pattern will not be loaded.
- ! Ensure that the tactor address lookup table (*tactor.lut*), provided with your Tactile Display, is in the pattern directory.
- ! Ensure that valid display ids are used in a pattern, or the pattern will not play.
- ! Ensure that valid body part ids are used in a pattern, or the pattern will not play.
- ! If the number of actions in a pattern or a merged pattern exceeds 50 then only the first 50 actions will be played.

### 4.1. Create a Tactile Pattern

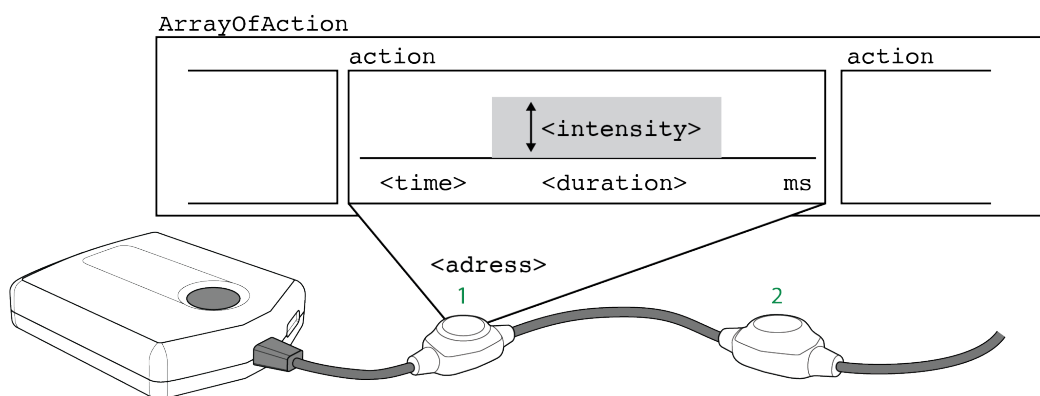
A tactile pattern file can be used to describe patterns in two different ways:

1. tactors are controlled directly by using their tactor numbers/addresses;
2. tactors are controlled indirectly by indicating a location on the skin where a vibration should be felt by using the body surface coordinate system (see [5 Body Surface Coordinate System](#)).

Both options are described below.

A tactile pattern consists of an "ArrayOfAction", that may contain multiple actions. An action describes a single vibration. It describes when and where the vibration occurs, how long it lasts and what intensity it has. Actions may be overlapping in time and space. An "ArrayOfAction" may contain the following global variables:

- <displayID> Unsigned integer indicating the tactile display id the pattern should be played on. Holds for all actions, unless specified differently inside an action.
- <bodyPart> Enumerate that indicates the body part the pattern should be played on [UNSPECIFIED; TORSO; LEFTUPPERARM; RIGHTUPPERARM; LEFTLOWERARM; RIGHTLOWERARM; LEFTHAND; RIGHTHAND; LEFTUPPERLEG; RIGHTUPPERLEG; LEFTLOWERLEG; RIGHTLOWERLEG; LEFTFOOT; RIGHTFOOT; HEAD]. Holds for all actions, unless specified differently inside an action. Shall be omitted when <Address> is specified in actions.



Within an action you can modify:

- <Time> Offset (in ms) before the action is executed. Range 0-60000 ms; resolution 5 ms. When it is zero, the tactor is activated immediately.
- <Address> The tactor number, counted from the control module (1-32). Cannot be used together with either displayID or bodyPart. Shall be omitted when either <circumferenceCoor> and/or <verticalCoor> are/is specified.

<displayID>	(optional) See above. Overwrites any global <displayID>.
<bodyPart>	(optional) See above. Overwrites any global <bodyPart>. Shall be omitted when <Address> is specified.
<circumferenceCoor>	Integer [0...1023] indicating the circumference coordinate. Shall be omitted when <Address> is specified.
<verticalCoor>	Integer indicating the vertical coordinate. Shall be omitted when <Address> is specified.
<Intensity>	The logarithmic stimulation intensity (0-15). Zero is the lowest setting and does not mean off. 15 means full power.
<Duration>	The duration of the vibration (in ms). Range 0-60000 ms; resolution 5 ms. Zero indicates that the tactor should be switched off.

## 4.2. How to format pattern files

The predefined tactile patterns that can be played using *Elitac HIDCom* have to be defined in files that use the XML format. All patterns that belong together should be grouped in a single directory. You can create the XML files manually in a text editor. The format of the pattern should be as follows.

**The example below will switch tactor number 7 at intensity level 3 for 500 ms:**

```
<?xml version="1.0" encoding="utf-8"?>
<ArrayOfAction Priority="Now" >
  <Action>
    <Time>3500</Time>
    <Address>7</Address>
    <Intensity>3</Intensity>
    <Duration>500</Duration>
  </Action>
</ArrayOfAction>
```

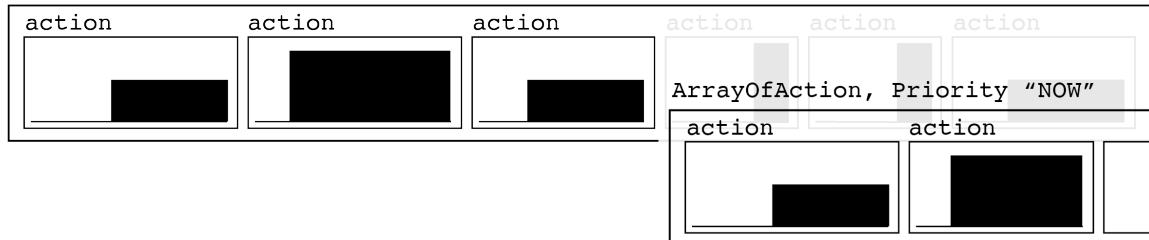
**The example below will create a vibration of intensity 3 at display 1 on the torso at body coordinate (0,512):**

```
<?xml version="1.0" encoding="utf-8"?>
<ArrayOfAction Priority="Now" >
  <displayID>1</displayID>
  <bodyPart>TORSO</bodyPart>
  <Action>
    <Time>3500</Time>
    <circumferenceCoor>0</circumferenceCoor>
    <verticalCoor>512</verticalCoor>
    <Intensity>3</Intensity>
    <Duration>500</Duration>
  </Action>
</ArrayOfAction>
```

### 4.3. Priority of tactile patterns between ArrayOfAction

The example has a priority property set that controls the behavior of the system when you try to play two patterns simultaneously. The priority keyword can be found after the `ArrayOfAction`. It can have one of the following four values: "Now", "Whenever", "Whatever" and "Undefined" (quotes are required). When omitted the pattern will have priority "Undefined".

ArrayOfAction, Priority "Whenever"



The default behavior of the system is as follows:

Priority of ...

... pattern to play >	Now	Whenever	Whatever	Undefined
... pattern being played v				
Now	Play both simultaneously	Wait until the current pattern has finished, then play new pattern	Do not play new pattern	Play both simultaneously
Whenever	Stop current pattern and play new pattern	Wait until the current pattern has finished, then play new pattern	Do not play new pattern	Play both simultaneously
Whatever	Stop current pattern and play new pattern	Wait until the current pattern has finished, then play new pattern	Do not play new pattern	Play both simultaneously
Undefined	Play both simultaneously	Wait until the current pattern has finished, then play new pattern	Do not play new pattern	Play both simultaneously

## 5. Body Surface Coordinate System

The Body Surface Coordinate system enables developers to indicate where on the skin a vibration should be felt, instead of using only tactor addresses.

This is a body invariant coordinate system referenced by anatomical landmarks (such as top of hip point, joints). For each body part a cylindrical coordinate system is defined. When the body is in anatomical position, the vertical axis points downwards and starts and ends at anatomical landmarks.

The start is defined as 0 and the end as 1023. Integers are used to indicate vertical coordinates which may extend beyond the range [0...1023]. A circumference axis is defined that runs in a clockwise fashion around the body part, starting and ending at the vertical axis. Its integer value ranges from [0...1024). Together the axes form a righthanded coordinate system (See [Figure 5](#), [Figure 6](#) and [Figure 7](#) for examples). A location on the skin is uniquely defined by the name of the body part and its circumference and vertical coordinate.

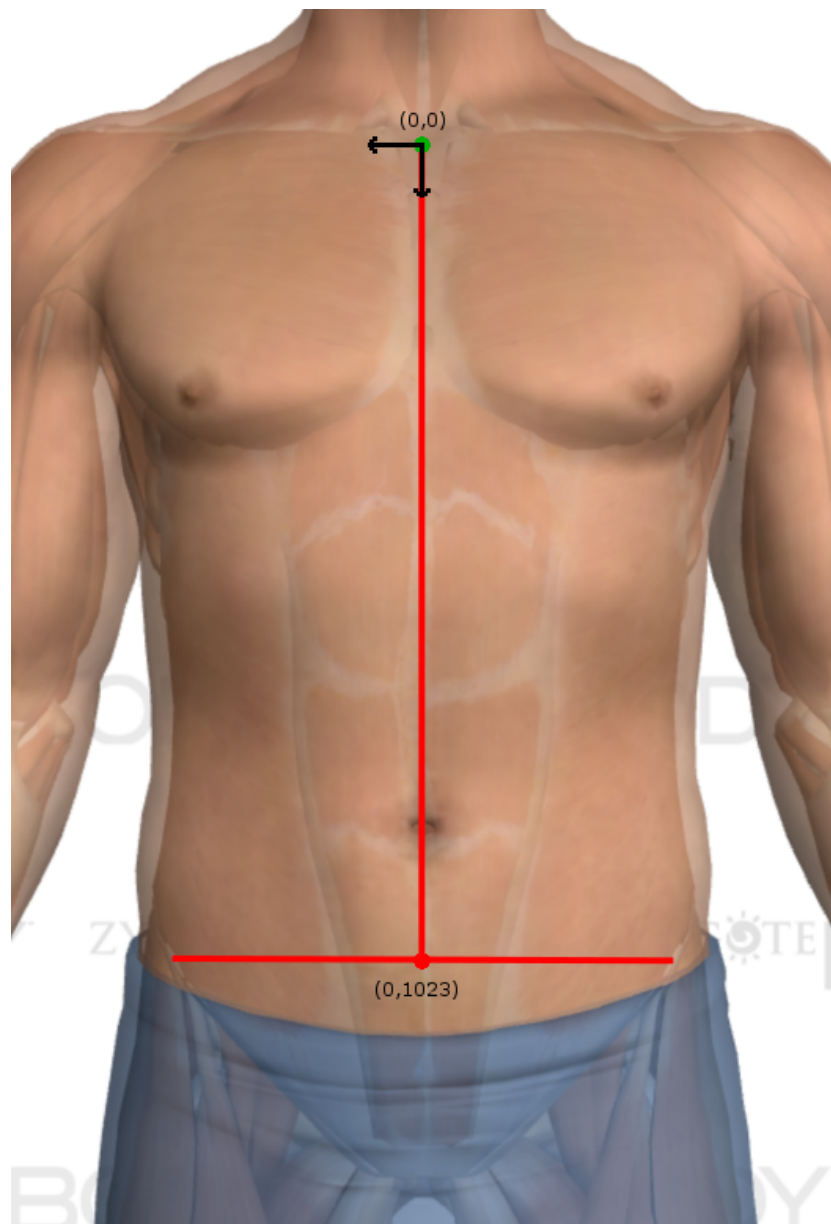


Figure 5 – Example of the coordinate system on the torso. The green dot represents the origins of surface coordinate system. Only the vertical axis is shown. The red dot indicates the end of the vertical axis. The circumference axis runs around the body part in a clockwise fashion. Dots are located at anatomical landmarks.

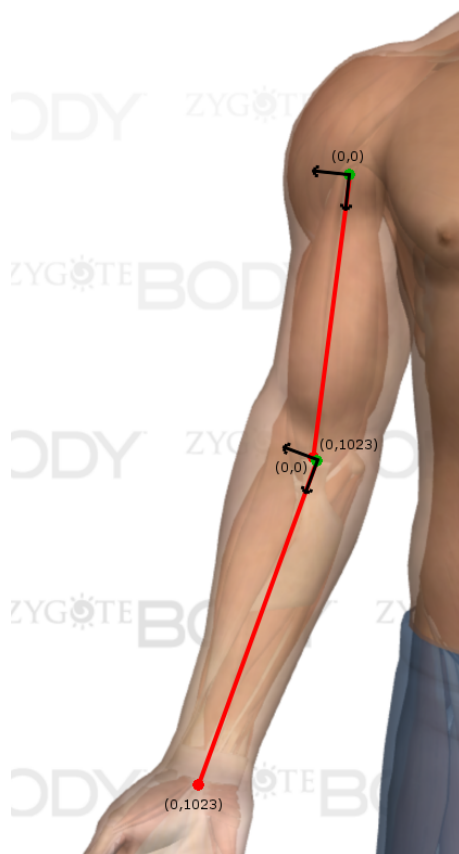


Figure 6 – Example of the coordinate system on upper and lower right arm. Green dots represent origins of the surface coordinate systems. Only the vertical axis is shown. The red dots indicate the ends of the vertical axes. The circumference axis runs around the body part in a clockwise fashion. Dots are located at anatomical landmarks.

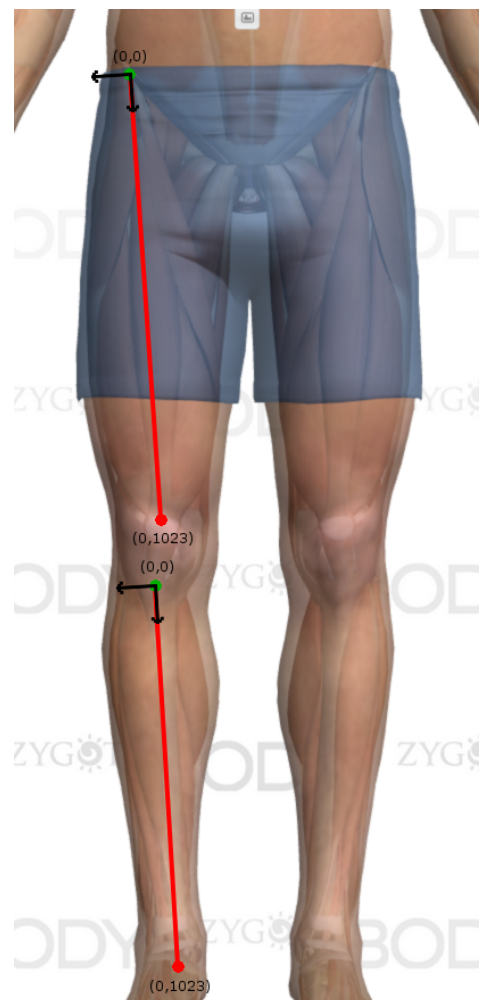


Figure 7 – Example of the coordinate system on upper and lower right leg. Green dots represent origins of the surface coordinate systems. Only the vertical axis is shown. The red dots indicate the ends of the vertical axes. The circumference axis runs around the body part in a clockwise fashion. Dots are located at anatomical landmarks.

## 6. Tactor Constellations and Tactile Displays

A tactor constellation is a spatial set of tactors that together form a functional tactile display. The constellation has a unique display id and the location of its tactors on the skin is defined in body surface coordinates.

In addition, a tactile display contains the relation between the tactors; the tactors in a display can either form a set of distinct points, a line or a triangle mesh. When you want to create a vibration at a given body surface coordinate, you select the correct display and the Elitac hardware will use tactor interpolation (for point displays nearest interpolation is used) to create a virtual vibration at the desired body surface coordinate (see Figure 8). If you want to know how to do this, then check [3 Controlling the Tactile Display from your own application](#).

The tactor constellation are programmed into the Elitac firmware. Consult the user manual of your specific hardware set to check which display are available to you. Repositioning the tactors in the shirt would probably require an adaptation of the tactor constellation in the firmware. If you want to change the layout or to define your own displays, please contact Elitac.

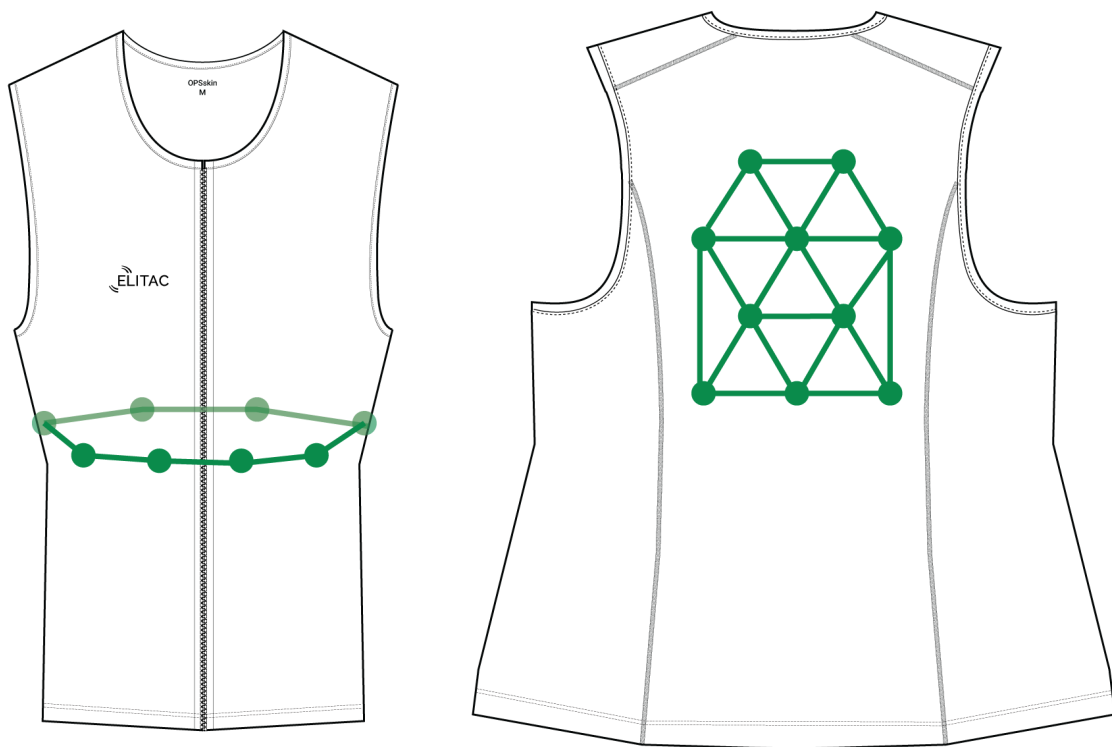


Figure 8 – On the left an example of a tactor constellation that forms a line display. On the right an example of a tactor constellation that forms a mesh (triangle) display. Locations of vibrations that are displayed on these displays are defined in body surface coordinates. The tactile hardware will use tactile interpolation to create virtual vibrations at these locations.



## 7. Troubleshooting

If any problems occur during installation or when using the HIDCOM software, use the following troubleshooting list to find a solution for the problem.

If your problem cannot be solved using this list contact Elitac for support [support@elitac.nl](mailto:support@elitac.nl).

### 7.1. Elitac HIDCom Software

---

**When I try to connect through USB the Elitac HIDCom is not showing the device or I cannot connect to it.**

1. Attach the USB connector correctly to the device and computer.
  2. Switch the device on and disable Bluetooth in the Settings of *Elitac HIDCom*
- 

### 7.2. Patterns

---

**When I send !PlayPattern,<patternName> to the *Elitac HIDCom* application it returns -1 and the tactors are not activated.**

1. Check if the pattern name is correct (without xml extension).
  2. Reload the correct pattern directory in *Elitac HIDCom*.
  3. Check if the corresponding pattern file exist in the pattern directory.
  4. Format the tactile pattern correctly following the guides [How to Format pattern files](#).
- 

**When I send !PlayPattern,<patternName> to *Elitac HIDCom* software application it returns its duration, but the tactors are not activated.**

1. Check if all cables are still intact, otherwise contact Elitac for support [support@elitac.nl](mailto:support@elitac.nl).
  2. Check if the connection between the Tactile Display and the computer is still up, and otherwise reactivate a connection through USB or Bluetooth®
- 

**I have just changed a pattern file, but when I play the pattern it plays the old version of the pattern.**

1. Open Elitac HIDCom settings. In pattern folder, click on "Open" to reload the pattern directory, to load the new pattern. [4.1 Install Elitac HIDCom](#)
- 

**I have just changed the tactor lookup table, but when I play a pattern the new tactor assignment does not seem to take effect.**

1. Reload the pattern directory in *Elitac HIDCom*.
  2. Check if the corresponding tactor lookup table file exists in the pattern directory.
  3. Format the tactor lookup table file correctly [4.4 How to create tactile patterns?](#).
- 

## 8. Technical support

If any problems occur during use, use the troubleshooting list to find a solution for the problem.

Technical support is available at cost price. SLA and/or custom work can be agreed separately, upon request.

For technical support contact Bram Bicknese, [support@elitac.nl](mailto:support@elitac.nl)

## 9. Register

Create tactile patterns .....	11
Connecting .....	5
Controlling .....	8
Coordinate system .....	14
Format pattern files .....	12
Installation .....	5
Pattern priority .....	13
Tactile display .....	4, 16
Tactor constellations .....	16
Technical support .....	17
Troubleshooting .....	17

## 10. Glossary

### Action

An action describes what a single tactor should do within a tactile pattern.

### (Tactor) Address

Parameter in tactile pattern; The tactor number, counted from the control module (0-32).

### ArrayOfAction

A tactile pattern, consisting of multiple actions.

### Control module

The main control unit of the Tactile Display. This module contains the processor as well as the battery.

### C-Type USB-plug

A C-Type USB-plug that connects the USB-cable to the Control Module

### Duration

Parameter in tactile pattern; Duration of the vibration.

### Elitac HIDCom

Software that can be used to play (custom) predefined tactile patterns.

### Firmware

Software installed on the Tactile Display

### Intensity

Parameter in tactile pattern; The logarithmic stimulation intensity (0-15).

### LED; see indicator light

### Manual

Extensive explanation on the (first time) use of the tactile display as well as more advanced functionalities.

### Tactile Display

A system that uses the skin to display information using localized vibrations comprising of a control module, tactile string and textiles.

### Tactile Pattern

A pattern on multiple tactors described as an ArrayOfAction

### Tactor

A small component that transmits vibrations.

### Tactor constellation

A spatial set of related tactors.

### Time

Parameter in Tactile pattern; Offset (in ms) before the action is executed. When it is zero, the tactor is activated immediately.

### USB-Cable

Data cable used to charge the Tactile Display, as well as to read/write the firmware using the *Elitac HIDCom* Software.