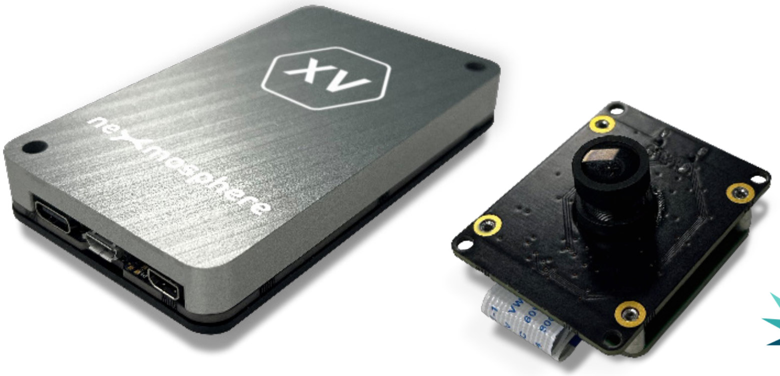
Table of content



MOTION GESTURES
the next wave of interaction

1. General

Nexmosphere's XV HandGesture sensor can track the position of a person's hand and detect gestures such as pointing, swiping and thumbs up. This functionality can be used to make touchless user interfaces for public spaces. This document provides explanation of the available functionalities and instructions on how to install and integrate the sensor into your digital signage installation.

The information in this document is created for users who are familiar with the Nexmosphere API and are able to control a basic setup with a Nexmosphere API controller. If this is not the case yet, please read the general documentation on the Nexmosphere serial API first.

2. Product overview

The XV HandGesture sensor consists out of 2 items:

XV-H40 HandGesture sensor

S-CLA05 Camera module

	XV-H40
HandGesture detection	✓
Position tracking	✓
Activation zone detection	✓
HDMI output	✓

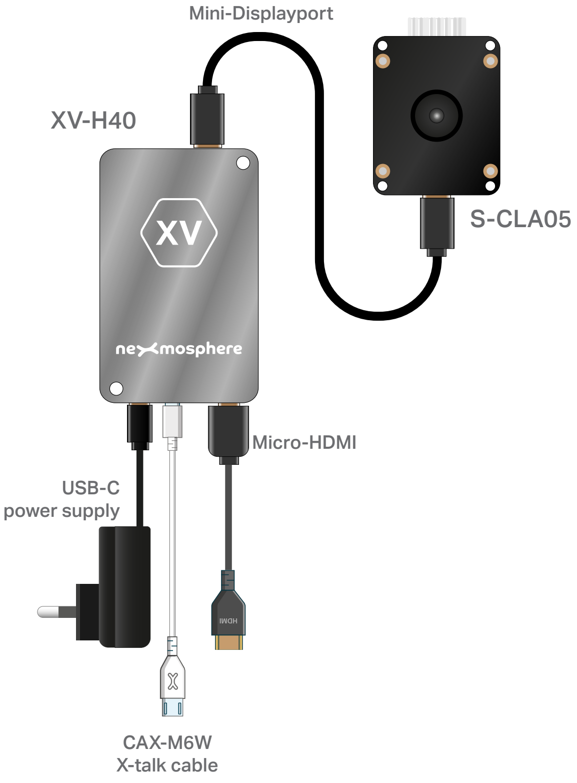
	S-CLA05
Max detection distance	180cm
Field of View	90°

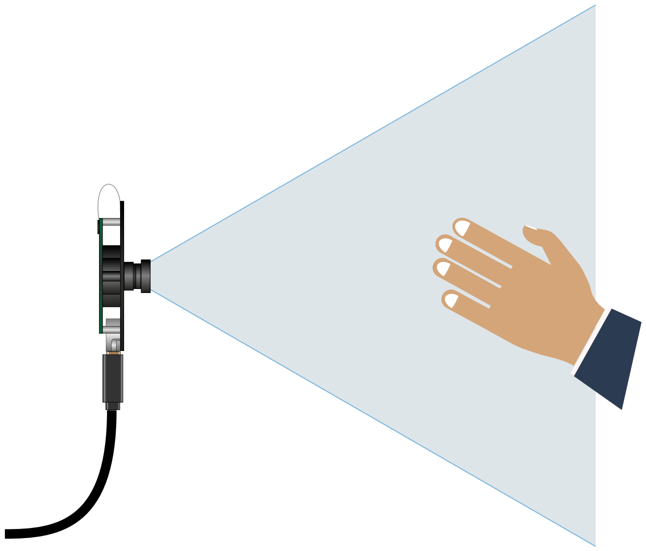
To create a functioning HandGesture sensor, the S-CLA05 camera module needs to be connected to the XV-H40 module with a Mini-Displayport cable (included). The XV-H40 module can then be powered with the provided USB-C power supply. The XV-H40 is connected to one of Nexmosphere's Xperience controllers using a CAX-M6W X-talk cable.

The XV-H40 sensor analyses the video feed captured by the S-CLA05 camera module utilizing the AI algorithm from Motion Gestures. This algorithm is embedded into the sensor and does not need to be purchased or licensed separately.

The sensor will detect hand gestures such as pointing, swiping and thumbs up. Next to this, it can track the position of a person's hand and send out a trigger when it enters an Activation zone.

MOTION GESTURES
the next wave of interaction





3. Functionality and API commands

The XV HandGesture sensor provides the following functionalities:

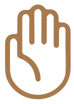
1. **HandGesture detection** - detects gestures, direction, orientation (front/back) and left/right hand
2. **Position tracking** - tracks the position and rotation of a person's hand
3. **Activation zones** - define customizable trigger zones and trigger when a hand enters
4. **Sensor configuration and initialization** - adjust the sensor's behavior and output
5. **HDMI output** - provides a video feed of the camera's FoV and hand tracking
6. **Focus mode** - helps to optimize the focus of the camera's lens
7. **USB updates** - to customize several aspects of the sensor's HDMI output
8. **Monitoring** - monitor the sensor's performance

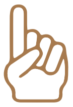
The following sections will cover each of these functionalities in detail. Please note that for each API example in this document, X-talk interface address 001 is used (X001). When the sensor is connected to another X-talk channel, replace the "001" with the applicable X-talk address.

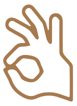
3.1 - HandGesture detection

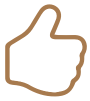
When a hand gesture is detected, an API message is triggered. Per default, this API message indicates the detected gesture and the direction of a person's hand. This API message has the following format:

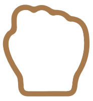
X001B[GEST=OPENPALM:***]	**** = direction* **up**, **down**, **left**, **right**, **center**
X001B[GEST=POINT:***]	**** = direction* **up**, **down**, **left**, **right**, **center**
X001B[GEST=OK:***]	**** = direction* **up**, **down**, **left**, **right**, **center**
X001B[GEST=THUMB:***]	**** = direction* **up**, **down**, **left**, **right**, **center**
X001B[GEST=FIST:***]	**** = direction* **up**, **down**, **left**, **right**, **center**
X001B[GEST=V:***]	**** = direction* **up**, **down**, **left**, **right**, **center**
X001B[GEST=SWIPE:***]	**** = direction* **left**, **right**
X001B[GEST=TAP]	
X001B[GEST=GRAB]	
X001B[GEST=NOHAND]	
X001B[GEST=NOGESTURE]	*

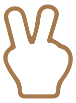

open palm

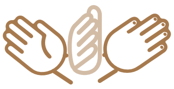

point

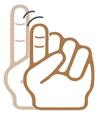

OK

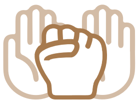

thumb


fist


V


swipe


tap


grab

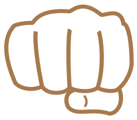
Gesture example API message

Point gesture detected with direction right
X001B[GEST=POINT:RIGHT]

Thumb gesture detected with direction up
X001B[GEST=THUMB:UP]

Swipe gesture detected with direction left
X001B[GEST=SWIPE:LEFT]

Fist gesture detected with direction center
X001B[GEST=FIST:CENTER]



Next to detecting a gesture and the direction, the sensor can detect if it is a **Left** or **Right** hand and whether the **Front** or **Back** of the hand is facing the sensor. Furthermore, the sensor can assign a **Hand ID** to the detected hand, which serves as an identification when two hands are detected simultaneously. In order to receive all parameters in the API message, the following setting must be send:

X001S[5:9]	Set output mode to Gesture with Direction, Left/Right hand, Front/Back, Hand ID and L/R hand

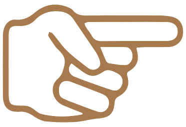
This API message has the following format:

X001B[#:@:%:GEST=OPENPALM:***]	#=Hand ID 1 or 2,	@= left/right L or R,	%= front/back F or B,	***= direction up, down, left, right, center
X001B[#:@:%:GEST=POINT:***]	#=Hand ID 1 or 2,	@= left/right L or R,	%= front/back F or B,	***= direction up, down, left, right, center
X001B[#:@:%:GEST=OK:***]	#=Hand ID 1 or 2,	@= left/right L or R,	%= front/back F or B,	***= direction up, down, left, right, center
X001B[#:@:%:GEST=THUMB:***]	#=Hand ID 1 or 2,	@= left/right L or R,	%= front/back F or B,	***= direction up, down, left, right, center
X001B[#:@:%:GEST=FIST:***]	#=Hand ID 1 or 2,	@= left/right L or R,	%= front/back F or B,	***= direction up, down, left, right, center
X001B[#:@:%:GEST=V:***]	#=Hand ID 1 or 2,	@= left/right L or R,	%= front/back F or B,	***= direction up, down, left, right, center
X001B[#:@:%:GEST=SWIPE:***]	#=Hand ID 1 or 2,	@= left/right L or R,	%= front/back F or B,	***= direction left, right
X001B[#:@:%:GEST=TAP]	#=Hand ID 1 or 2,	@= left/right L or R,	%= front/back F or B	
X001B[#:@:%:GEST=GRAB]	#=Hand ID 1 or 2,	@= left/right L or R,	%= front/back F or B	
X001B[GEST=NOHAND]				
X001B[GEST=NOGESTURE]				

Gesture example API message

Point gesture detected with direction right, Hand ID 1, Right hand, Front side

X001B[1:R:F:GEST=POINT:RIGHT]



Thumb gesture detected with direction up, Hand ID 2, Left hand, Back side

X001B[2:L:B:GEST=THUMB:UP]

The **output format** of the API messages can also be set to send each parameter in an individual command:

X001S[6:2]	Set output format of Hand Gesture detection to separate commands for each parameter

When the output mode is set to send all parameters (as in the examples on this page), 5 separate API messages will be send consecutively in the following format:

X001B[ID=#]	# = Hand ID 1 or 2
X001B[HAND=@]	@ = left/right LEFT or RIGHT
X001B[SIDE=%]	% = side FRONT or BACK
X001B[GESTURE=^^^]	^^^ = gesture OPENPALM, POINT, OK, THUMB, FIST, V, SWIPE, TAP, GRAB, NOHAND, NOGESTURE
X001B[DIRECTION=***]	*** = direction up, down, left, right, centre

Gesture example API message

Point gesture detected with direction right, Hand ID 1, Right hand, Front side

X001B[ID=1]
X001B[HAND=RIGHT]
X001B[SIDE=FRONT]
X001B[GESTURE=POINT]
X001B[DIRECTION=LEFT]

When implementing gesture triggers, consider the following:

- The sensor can be set to different output modes which have several combination of the parameters explained above. For more information, please see page 22, setting 5.
- Detection can be activated and deactivated for each gesture. For more information, please see page 22 setting 20-28.
- For more detailed information on the gestures itself, please see page 19-20.

Hand Gesture Data requests

The currently detected Hand Gesture can also be requested at any time by sending one of the following API commands:

`X001B[GEST1?]`	Request Gesture only
`X001B[GEST2?]`	Request Gesture with Direction
`X001B[GEST3?]`	Request Gesture with Direction and Front/Back
`X001B[GEST4?]`	Request Gesture with Direction and Hand ID
`X001B[GEST5?]`	Request Gesture with Direction and Left/Right
`X001B[GEST6?]`	Request Gesture with Direction, Front/Back and Hand ID
`X001B[GEST7?]`	Request Gesture with Direction, Front/Back and Left/Right
`X001B[GEST8?]`	Request Gesture with Direction, Hand ID and Left/Right
`X001B[GEST9?]`	Request Gesture with Direction, Front/Back, Hand ID and Left/Right

The reply is identical to the triggered API messages such as the examples listed on the previous pages.

When requesting the detected gesture for a specific Hand ID, the ID number can be added to the command with a #.

Example Hand Gesture Data request

Request the Gesture with Direction of Hand ID 1 `X001B[GEST2?#1]`

3.2 - Position tracking

The XV HandGesture sensor can also track the position, size and angle of a person's hand. The feature can be enabled by sending the following setting command:

`X001S[12:3]`	Set output interval of hand position information to 500mS

This API message has the following format:

X001B[#:XXX:YYY:SSS:AAA**]**

#=Hand ID **1** *or* **2***,* *XXX= X-coordinate* **0** *to* **100** *(%),* *YYY= Y-coordinate* **0** *to* **100** *(%),* *SSS= size* **0** *to* **100** *(%),* *AAA= angle* **0** *to* **360** *(°)*

Position example API message

Hand ID 1, position X= 50% of screen, Y=70% of screen, size = 12% of screen, angle is 30 degrees.

X001B[1:050:070:012: 030]

Hand ID 2, position X= 35% of screen, Y=2% of screen, size = 8% of screen, angle is 332 degrees.

X001B[2:035:002:08:332]

The output format of the API commands can also be set to send each parameter in an individual API message:

`X001S[13:2]`	Set output format of Position tracking to separate commands for each parameter

When the output mode is set to send all parameters (as in the examples on this page), 5 separate API messages will be send consecutively in the following format:

X001B[ID=#**]**	*#=Hand ID* **1** *or* **2**
X001B[X=XXX**]**	*XXX= X-coordinate* **0** *to* **100** *(%)*
X001B[Y=YYY**]**	*YYY= Y-coordinate* **0** *to* **100** *(%)*
X001B[S=SSS**]**	*SSS= size* **0** *to* **100** *(%)*
X001B[A=AAA**]**	*AAA= angle* **0** *to* **360** *(°)*

Position example API message

Hand ID 1, position X= 50% of screen, Y=70% of screen, size = 12% of screen, angle is 30 degrees.

X001B[ID=1]
X001B[X=050]
X001B[Y=070]
X001B[S=012]
X001B[A=030]

When implementing position tracking, consider the following:

- The X and Y coordinates are provided in % of the camera's Field of View. The top left of the screen is the origin (0,0).
- The size is indicated in % of the height of the camera's Field of View.
  See page 7 for a visual explanation of the X and Y coordinates as well as the hand size.
- Per default, the reference point for the X and Y coordinate on the detected hand is the tip of the Index finger. This can be adjusted via a setting command. For more information please see page 21, setting 16.
- The sensor can be set to different output modes which have several combination of the parameters explained above. For more information, please see page 22, setting 11.
- The output interval of the sensor can be adjusted as well, for more information please see page 21, setting 12. As the shortest available interval is 250mS, the sensor is not suitable for high speed position tracking (for example to control a mouse cursor). For applications in which a specific position of a hand should cause a trigger, we recommend using the Activation Zones feature, explained in section 3.3, page 8.

Position Data requests

The currently detected Position can also be requested at any time by sending one of the following API commands:

`X001B[XY?]`	Request X and Y coordinates of the detected hand
`X001B[SIZE?]`	Request the Size of the detected hand
`X001B[ANGLE?]`	Request the Angle of the detected hand
`X001B[POSITION?]`	Request X, Y coordinates, Size and Angle of the detected hand

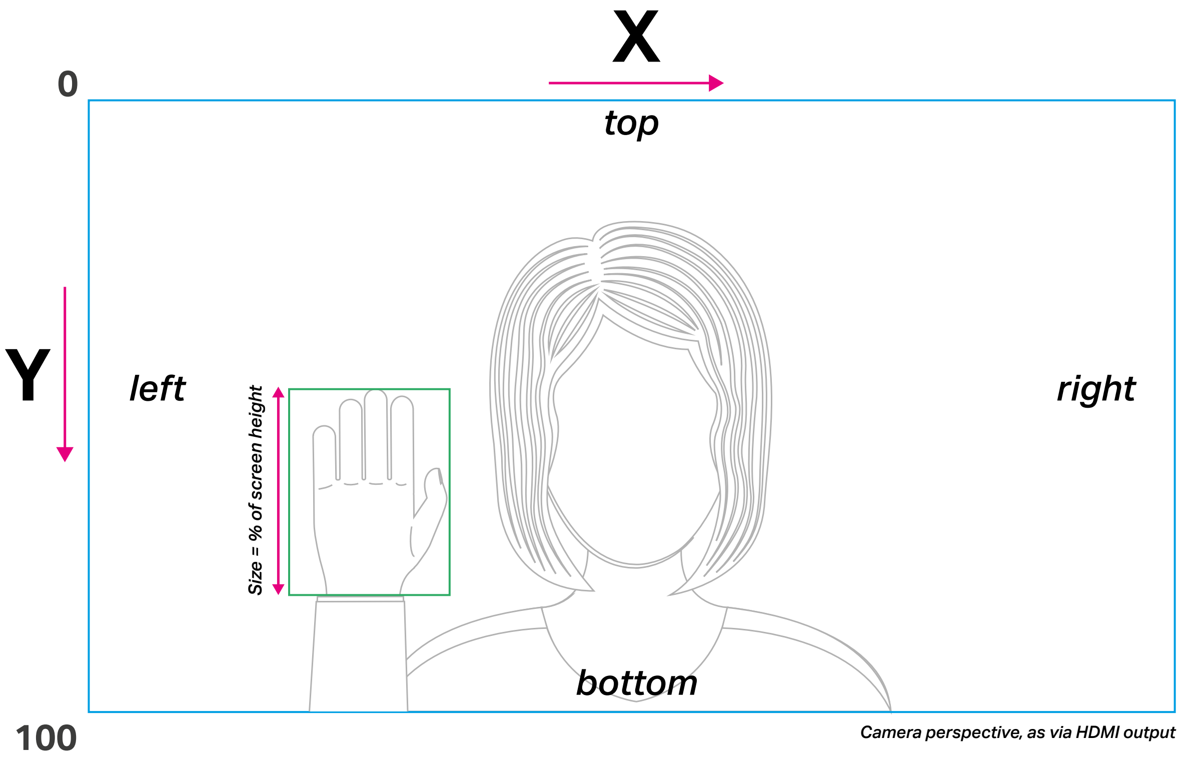
The reply is identical to the triggered API messages such as the examples listed on the previous pages.

When requesting the position for a specific Hand ID, the ID number can be added to the command with a #.

Example Position Data request

Request the Position of Hand ID 2

```
X001B[POSITION?#2]
```



Camera perspective, as via HDMI output

3.3 - Activation zones

The XV HandGesture sensor has user-definable zones which can activate a trigger when the detected hand enters that zone. There are 9 activation zones available of which each the position and size can be set. The API command to define an Activation Zone is:

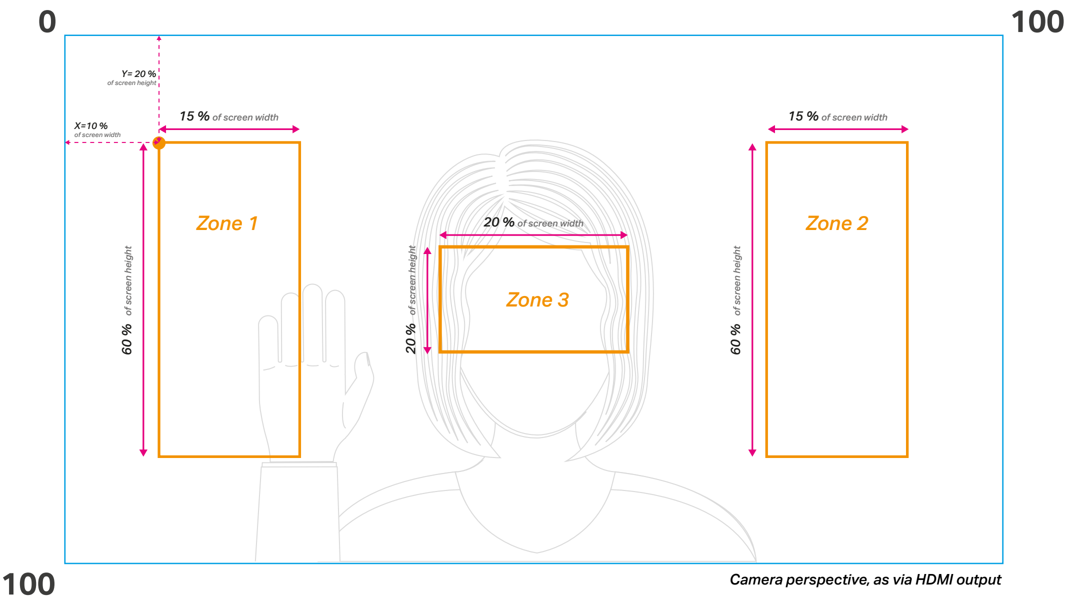
```
X001B[ZONE*=XXX,YYY,WWW,HHH]
```

Set position and size of Activation zone

** = Activation Zone*	**1** *to* **9** *(nr)*
XXX= X-coordinate	**0** *to* **100** *(%)*
YYY= Y-coordinate	**0** *to* **100** *(%)*
WWW= width	**0** *to* **100** *(%)*
HHH= height	**0** *to* **100** *(%)*

Set Activation zone example command

Set Act. zone 1 to X= 10%, Y=20%, Width=15%, Height=60%	`X001B[ZONE1=010,020,015,060]`
Set Act. zone 2 to X= 75%, Y=20%, Width=15%, Height=60%	`X001B[ZONE2=075,020,015,060]`
Set Act. zone 3 to X= 40%, Y=40%, Width=20%, Height=20%	`X001B[ZONE3=040,040,020,020]`



Camera perspective, as via HDMI output

When the HDMI output mode is set to DEBUG (see section 3.5, page 13), the Activation zones can be visualized on top of the camera feed. In order to activate the visualization of the Activation zones or update them after the parameters have been changed, the following API command must be send:

```
X001B[SENDZONES!]
```

Update the visualization of the Activation zones in Debug mode

When a detected hand enters one of the Activation zones, the following API message is send:

X001B[#:ZONE*:ENTER]

#=Hand ID **1** *or* **2** **=Activation Zone* **1** *to* **9**

When a detected hand exits one of the Activation zones, the following API message is send:

X001B[#:ZONE*:EXIT]

#=Hand ID **1** *or* **2** **=Activation Zone* **1** *to* **9**

Activation zone example messages

Detected hand enters Activation zone 3
X001B[1:ZONE3:ENTER]

Detected hand exits Activation zone 9
X001B[1:ZONE9:EXIT]

Next to the size and position, several other parameters can be adjusted for each Activation zone:

```
X001B[ZONE*DELAY=TTTT]
```

Set the time a hand needs to be in the Activation zone to trigger

** = Activation Zone*	**1** *to* **9** *(nr)*	
TTTT= time	**0** *to* **5000** *(mS)*	*Default = 100mS*

```
X001B[ZONE*ANGLE=AA1,AA2]
```

Set the angle range in which a hand needs to be to cause a trigger

** = Activation Zone*	**1** *to* **9** *(nr)*	
AA1= angle 1	**0** *to* **360** *(°)*	*Default = 0 °*
AA2= angle 2	**0** *to* **360** *(°)*	*Default = 360 °*

```
X001B[ZONE*HANDSIZE=SS]
```

Set the minimum size which a hand needs to be to cause a trigger

** = Activation Zone*	**1** *to* **9** *(nr)*	
SS= size	**0** *to* **80** *(% of FoV height)*	*Default =0%*

Activation zone parameters example commands

Set the time a hand needs to be in Activation zone 1 to cause a trigger to 500mS.

```
X001B[ZONE1DELAY=0500]
```

Set the angle range a hand needs to be in for Activation zone 5 to 150-210

```
X001B[ZONE5ANGLE=150,210]
```

Set the minimum hand size for Activation zone 9 to 20

```
X001B[ZONE1HANDSIZE=20]
```

The following control commands can be used to clear the Activation zone parameters. They will clear the size and position values, and will set the delay, angle and size back to the default values.

```
X001B[ZONE*=CLEAR]
```

Clear all parameters for a specific Activation Zone

** = Activation Zone* **1** *to* **9** *(nr)*

```
X001B[CLEARALLZONES]
```

Clear all paramers for all Activation Zones

When implementing Activation zones, consider the following:

- The X and Y coordinates are defined in % of the camera's Field of View. The top left of the screen is the origin (0,0).
- The width and height is defined in % of the camera's Field of View.
- The minimum handsize is defined in % of the height of the camera's Field of View.
- Per default, the Activation zone triggers when any hand enters. This can be adjusted to only trigger on detected gestures (in the Activation zone) via a setting command. For more information please see page 21, setting 15.
- Per default, the reference point for the X and Y coordinate on the detected hand is the tip of the Index finger. This reference point is used to determine whether a hand enters or exits an Activation Zone. The reference point can be adjusted via a setting command. For more information please see page 21, setting 16.
- The hysteresis of the Activation Zones can be adjusted via a setting command. For more information please see page 21, setting 17.

Activation zones Data request

The current Activation zone parameters can be requested at any time by sending he following API command:

`X001B[ZONES?]`	Request all parameters for all Activationzones

The replies are identical to the commands used to set the parameters:

X001B[ZONE*=XXX**,**YYY**,**WWW**,**HHH**]**	** = Activation Zone*	**1** *to* **9** *(nr)*
	XXX= X-coordinate	**0** *to* **100** *(%)*
	YYY= Y-coordinate	**0** *to* **100** *(%)*
	WWW= width	**0** *to* **100** *(%)*
	HHH= height	**0** *to* **100** *(%)*
X001B[ZONE*DELAY=TTT**]**	** = Activation Zone*	**1** *to* **9** *(nr)*
	TTTT= time	**0** *to* **5000** *(mS)*
X001B[ZONE*ANGLE=AA1**,**AA1**]**	** = Activation Zone*	**1** *to* **9** *(nr)*
	AA1= angle 1	**0** *to* **360** *(°)*
	AA2= angle 2	**0** *to* **360** *(°)*
X001B[ZONE*HANDSIZE=SS**]**	** = Activation Zone*	**1** *to* **9** *(nr)*
	SS= size	**0** *to* **80** *(%)*

3.4 - Sensor configuration and Initialization

The XV HandGesture sensor has multiple configuration commands available which determine the behaviour and functionality of the AI engine running on the sensor. In order for these commands to be initialized, the following API command must be send **after** one or multiple configuration commands have be send:

`X001B[START!]`	(Re)start AI engine (with new configuration commands)

Vice versa, it is also possible to place the sensor in an Idle state

`X001B[STOP!]`	Stop AI engine

The following configuration commands are available:

Per default the camera records in a resolution of 640x480. The resolution in which the camera records, also determines the resolution in which the HDMI connector outputs the video feed. This resolution can be adjusted to 1280x720 via an API command. Please note that the higher the camera resolution, the slower the hand detection will be, as it takes more time for the AI engine to analyse each frame. This API command has the following format:

`X001B[CAMRES=RRRR]`	Set the resolution of the camera capture and HDMI output		
	RRRR= resolution	**480** *or* **720**	*Default = 480*

For analysing dynamic gestures, the resolution must be kept at 480. When only using activation zones, the resolution can be set to 720. In this case the response and FPS will be slower, but the resolution of the video output on the HDMI will be higher. When adjusting the camera resolution, the resolution at which the AI engine analyses the video feed must be adjusted as well to match the value of the camera resolution. This API command has the following format:

`X001B[ALGRES=RRRR]`	Set the resolution of the AI engine		
	RRRR= resolution	**480** *or* **720**	*Default = 480*

Per default, the AI engine scans the entire FoV (Field of View) of the camera to detect a hand. This detection canvas can be decreased and set to a specific area of the FoV. A smaller canvas will result into faster hand detection. The feature can be used to only focus on specific areas or mimic optical zoom when combined with a higher camera resolution. This API command has the following format:

`X001B[CANVAS=XXX,YYY,WWW,HHH]`	Set the active detection canvas for the AI engine		
	XXX= X-coordinate	**0** *to* **100** *(%)*	*Default = 0*
	YYY= Y-coordinate	**0** *to* **100** *(%)*	*Default = 0*
	WWW= width	**0** *to* **100** *(%)*	*Default = 100*
	HHH= height	**0** *to* **100** *(%)*	*Default = 100*

See page 7 for an explanation and visualization of the definition of the coordinates, width and height.

The camera feed can be rotated 180 degrees. This is useful in cases where it is preferable to install the camera in an up-side-down position in which the cable protrudes from the top of the camera module instead of the bottom. This API command has the following format:

`X001B[CAMROT=180]`	Rotate the camera feed 180 degrees (cable on top)
`X001B[CAMROT=000]`	Default camera feed orientation (cable on bottom)

Per default, the sensor will only detect hands with a minimum size of 10% of the height of the sensor's Field of View. De facto, this determines the maximum distance at which a hand is detected (larger size = closer to sensor). This threshold can be adjusted with the following API command:

`X001B[MINSIZE=SS]`	Set the minimum size for a hand to be detected		
	SSS= size	**00** *to* **80** *(%)*	*Default =10*

The gestures "Swipe", "Tap" and "Grab" (see page 3), are dynamic gestures for which the AI engine on the sensor needs to analyse a sequence of video frames. The detection of these gestures is enabled by default, but can be disabled. The advantage of disabling the detection of dynamic gestures when not needed, is that it will make the detection of other gestures faster. The API commands to disable or enable dynamic gestures are:

`X001B[DYNAMIC=ON]`	Enable dynamic gestures (default)
`X001B[DYNAMIC=OFF]`	Disable dynamic gestures

The gestures "Open palm", "Point", "OK", "Thumb", "Fist and "V" (see page 3), are static gestures. The detection of these gestures is enabled by default, but can be disabled. The advantage of disabling the detection of static gestures when not needed, is that it will make the detection of dynamic gestures faster. The API commands to disable or enable static gestures are:

`X001B[STATIC=ON]`	Enable static gestures (default)
`X001B[STATIC=OFF]`	Disable static gestures

Per default, the sensor can detect 2 hands simultaneously. This can be adjusted to only detecting 1 hand at a time. This will make the sensor less flexible, as it keeps tracking the first hand it detects. However, it will make the hand detection faster. The corresponding API command have the following format:

`X001B[NRHANDS=1]`	Set AI engine to only detect 1 hand at a time
`X001B[NRHANDS=2]`	Set the AI engine to detect 2 hands simultaneously

Configuration Data requests

Several Data request are available regarding the sensor's configuration:

`X001B[CANVAS?]`	Request current active canvas
`X001B[NRHANDS?]`	Request the number of hands the sensors can detect simultaneously

The replies are identical to the commands used to set the parameters.

3.5 - HDMI output

The micro-HDMI connector on the XV HandGesture sensor can provide a real time video output of the camera feed. This output can be set to multiple modes. In order for these modes to be initialized the following command must be send **after** a new HDMI command has been send:

`X001B[START!]`	(Re)start AI engine (with new configuration commands)

The following HDMI-output commands are available:

`X001B[HDMI=DEBUG]`	Set the HDMI output to Debug mode. *Shows camera feed with additional debug information:* *Hand skeleton, tracking information, gesture detection, performance information*
`X001B[HDMI=FULL]`	Set the HDMI output to Full mode (Default mode). *Shows camera feed with gesture detection information and white skeleton*
`X001B[HDMI=CAMERA]`	Set the HDMI output to Camera mode. *Shows camera feed only*
`X001B[HDMI=ICON]`	Set the HDMI output to Icon mode. *Shows a (customizable) hand icon on the real-time position* *of the detected hand, no camera feed, customizable background image*
`X001B[HDMI=BLANK]`	Set the HDMI output to Blank mode. *Only shows customizable background image*
`X001B[HDMI=SKELETON:*]`	Set the HDMI output to Skeleton mode. *Only shows real-time hand skeleton, no camera feed,* *customizable background image*

** = Color scheme for Skeleton* **1** *to* **9** *(nr)*

1= white	*4= red*	*7= yellow*	*0= rainbow*
2 = grey	*5 = blue*	*8 = orange*	
3 = black	*6 = green*	*9 = purple*	

For modes with a background image (Icon, Blank and Skeleton), the background image can be customized via a USB-update procedure. This procedure is explained on page 15.

The hand icon image used in Icon mode can be customized via a USB-update procedure. This procedure is explained on page 15.

HDMI output Data request

The HDMI output mode can be requested at any time by sending he following API command:

`X001B[HDMI?]`	Request current HDMI mode

The reply will be identical to the command used to set the HDMI mode (listed above).

3.6 - Focus mode

During installation, the sensor can be set to a special Focus mode. This mode helps to optimize the focus of the camera lens to the person in front of the sensor. In order to start Focus mode, the following API commands must be sent to the sensor:

`X001B[STARTFOCUS!]`	Start Focus mode

When the sensor is running in Focus Mode, the HDMI output will show the live camera feed with the following information in the top left corner:

Focus = XXX
Best = XXX

The "Focus" value indicates the current focus level (higher is better) and the "Best" value indicates the highest focus level detected until then. The lens of the camera module can be rotated to adjust its focus point. In order to optimize focus, we recommend the following procedure:

1. Make sure a person stands in front of the sensor, at the distance where most people will stand when interacting
2. Ask the person to make a hand gesture that is used in the application, for example Point or Thumbs up.
   The person should try to stand as still as possible
3. Preferably, the background (what is happening behind the person) is as consistent as possible as well.
4. Start Focus mode by sending the start command listed above.
5. Rotate the camera lens to the left. If the Best value increases, keep rotating until it stops increasing and the "Focus" value (almost) matches the "Best" value. In case the Best value does not increase, but instead the "Focus" value decreases, go to step 6.
6. Rotate the camera lens to the right. If the Best value increases, keep rotating until it stops increasing and the "Focus" value (almost) matches the "Best" value.

The "Focus" and "Best" value are also provided via the following API message:

X001B[FOCUS:FFFF**,**BBBB**]**	*FFFF= Current focus value* **000** *to* **1000** *BBBB= Best focus value* **000** *to* **1000**

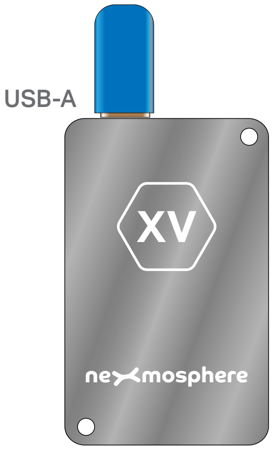
When the focus procedure is finished, the following command must be sent to place the sensor back in normal operation mode:

`X001B[START!]`	(Re)start AI engine (with new configuration commands)

3.7 - USB updates

The sensor offers several options to customize the look and feel of the video output of the sensor. The following options are available:

- Update the hand icon image used in Icon mode
- Update the background image used in Icon, Skeleton and Blank mode
- Update the image during start-up of the sensor
- Update the video shown when the sensor is in idle state (AI engine not running)



Update the hand icon image for Icon mode

The hand icon image can be customized by following the procedure below:

1. Create your custom hand icon image in the resolution 400x400,
   and place it in a 460x460 canvas, resulting in a 30px transparent border.
2. Save it as a .png with the following (exact) filename: Hand.png
3. Store the file on a USB stick
4. Insert the USB stick in the USB-A connector
5. Send a REBOOT! command (see page 16)
6. The sensor will automatically update the hand icon image with the customized image on the USB stick
7. The status of the upload procedure is indicated on your screen via the HDMI output connector.
8. After uploading has finished, the sensor will shut down. Unplug the USB stick and repower the sensor.

Update the background image for Icon, Skeleton and Blank mode

The background image can be customized by following the procedure below:

1. Create your custom background image in the following 2 resolutions, 640x480, 1280x720
2. Save them as a .jpg or .png with the following (exact) filenames:
   Background-480.jpg *or .png*
   Background-720.jpg *or .png*
3. Store the files on a USB stick
4. Insert the USB stick in the USB-A connector
5. Send a REBOOT! command (see page 16)
6. The sensor will automatically update the background images with the customized images on the USB stick
7. The status of the upload procedure is indicated on your screen via the HDMI output connector
8. After uploading has finished, the sensor will shut down. Unplug the USB stick and repower the sensor.

Update the start-up image of the sensor

The background image can be customized by following the procedure below:

1. Create your custom start-up image in the following 2 resolutions, 640x480, 1280x720
2. Save them as a .jpg or .png with the following (exact) filenames:
   Boot-480.jpg *or .png*
   Boot-720.jpg *or .png*
3. Follow step 3 to 7 listed in the sections above.

Update the idle video of the sensor

The video which is shown when the sensor is in idle mode can be customized by following the procedure below:

1. Create your custom videoe in the following 2 resolutions, 640x480, 1280x720
2. Save them as a .mp4 (exported with an H.264 codec)
   Idle-480.mp4
   Idle-720.mp4
3. Follow step 3 to 7 listed in the sections above.

3.8 - Monitoring and control

The sensor offers several features to monitor the status, performance and system health of the sensor. The following API messages are send automatically:

X001B[INFO=SENSOR READY]	Indicates when the sensor is ready for operation after a power cycle
X001B[WARNING=TEMP IS HIGH]	Warning message indicating that the sensor's temperature is high (65°C) *In this case, it is advisable to add additional airflow to the sensor installation*
X001B[WARNING=TEMP IS CRITICAL]	Warning message indicating that the sensor's temperature is critical (75°C) *Immediate action required to reduce the sensor's temperature*
X001B[INFO=TEMP IS OK]	Info message indicating that the sensor's temperature is back at an OK level *This message is only send in case the temperature has been critical or high*

When the 5V USB-C power supply is connected to the sensor, it will automatically start. It will take approximately **60 seconds** for the sensor get into operational mode. During operation, the sensor can be rebooted via the following API command:

`X001B[REBOOT!]`	Reboots the AI engine processors

After a reboot command, the sensor will restart into Idle mode. In order to set the sensor into operational mode, a START! command must be send when the sensor is ready rebooting (see page 11).

Monitoring Data request

The sensor's performance and system health can be checked with the following Data requests:

`X001B[FPS?]`	Request the FPS which the AI engine currently analyses (higher = faster)

The reply will be as follows:

X001B[FPS=FF]	*FF=FPS*	**00** *to* **99** *(Frames per Second)*

`X001B[TEMP?]`	Request the sensor's current temperature

The reply will be as follows:

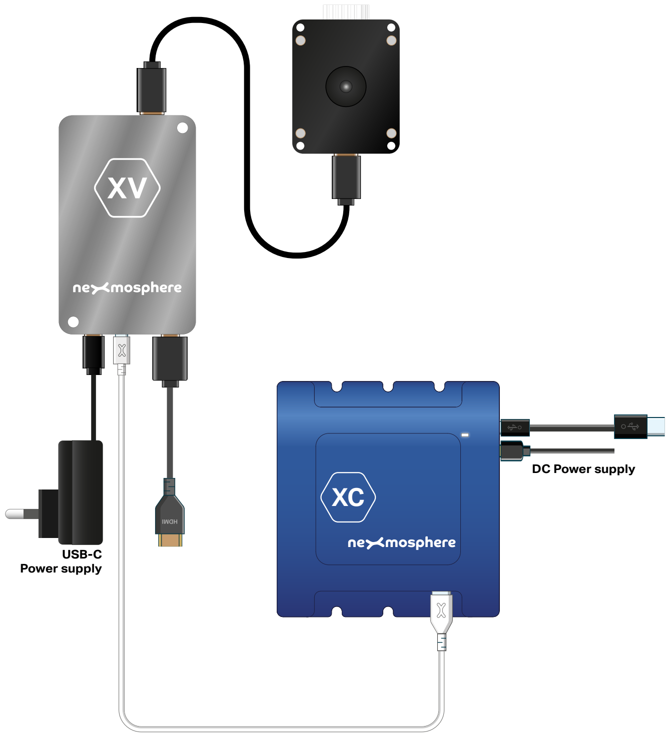
X001B[TEMP=TT°C]	*TT=Temperatue*	**00** *to* **99** *(degrees Celsius)*

4 - Installation requirements and guidelines

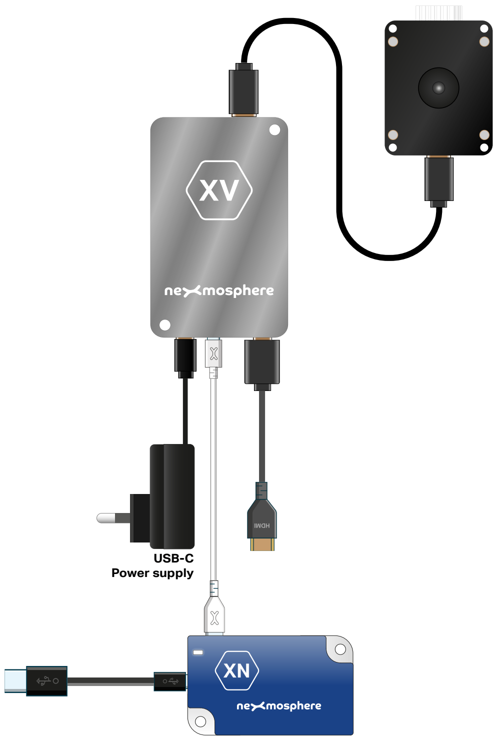
When integrating an XV HandGesture sensor into your digital signage installation, several installation requirements and guidelines need to be taken into account in order for the sensor to perform optimal and operate stable.

4.1 Connection Diagrams

The XV HandGestuer sensor can be connected to any X-talk interface and is therefore compatible with all Xperience controllers. Make sure the sensor is powered and connected to the X-talk interface before powering the Xperience controller. Otherwise, the sensor will not be recognized by the Xperience controller and no sensor output will be provided.



Example connection to XC Controller



Example connection to XN Controller

4.2 Hardware integration guidelines

When integrating an XV HandGesture sensor into your digital signage installation, several installation requirements and guidelines need to be taken into account in order for the sensor to perform optimal and operate stable.

Temperature regulation

The sensor is a 15watt device which dispenses heat through its full aluminium body. When installing the XV-H40 module, please make sure the environment has a (mild) airflow, allowing the device to stay below the absolute maximum operating temperate of **75**°C. Installing the sensor in a vertical position will help to create airflow.

Camera installation

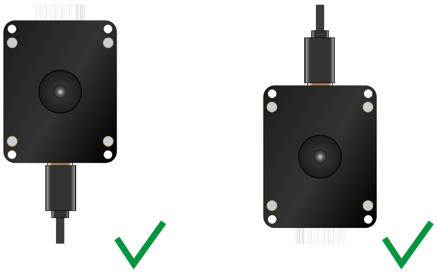
When installing the camera module, we recommend to take the following guidelines into account:

Camera orientation

The sensor can be installed in 2 orientations:

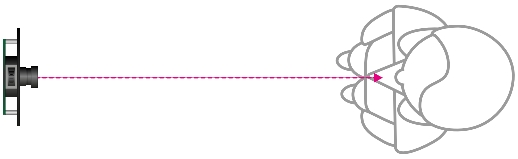
- Default orientation: DisplayPort connector on bottom
- 180° rotated: DisplayPort connector on top

When installing the sensor in the 180 degrees orientation, please make sure to send the corresponding API commands for the sensor setup (see page 11).



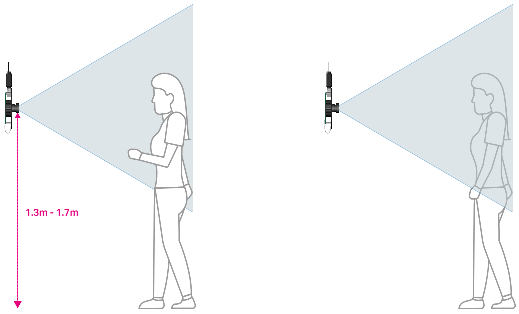
Horizontal positioning

Install the camera centered with the location of where the person interacting with the sensor will be standing. If this is not possible, we recommend installing the sensor as close as possible to the center, on the right hand side of the person standing in front of the sensor



Vertical positioning

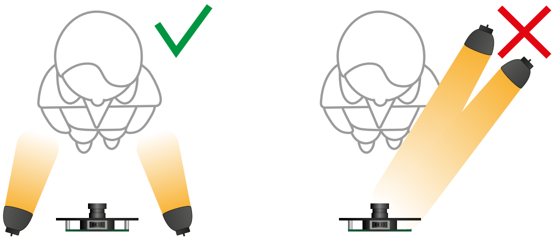
The sensor has a 90 degrees Field of View. We recommend to install the sensor at a height in which a slightly raised hand of the person in front of the sensor will cause the hand to enter the camera's field of view. Depending on the distance of the person in front of the display, typically the recommended installation height is between **1.3 - 1.7 meters.**



For the horizontal and vertical positioning of the camera, we recommend to always test with the HDMI feed and check if a person's hand is well within the sensor's field of view (also when making dynamic gestures, such as swipes).

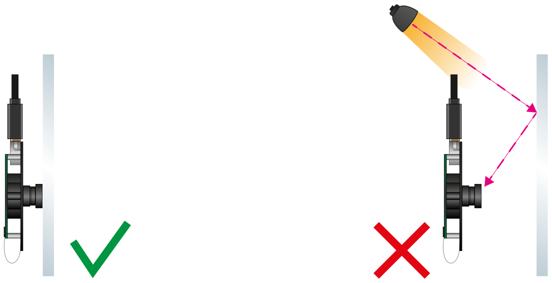
Light conditions

The sensor works optimally in even light conditions, meaning that there is no strong dark/light contrast in the sensor's Field of View. Especially strong light sources pointed directly towards the camera, such as a light bulb or sunshine through a window, can decrease the quality of the video feed and thereby the accuracy of the hand gesture detection.



Installation behind glass

The sensor works through glass or any clear transparent material through which there is full vision. When installing the camera behind glass, we recommend to install the lens as close as possible to the glass panel. This avoids light reflections in the camera's field of view.

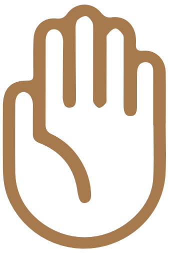


4.3 Guidelines for gesture detection

Hand gestures can be interpreted and executed in many different ways. Although the AI engine is trained to detect a variety of executions, the sensor will perform best when the following guidelines are taken into account when a gesture is executed.

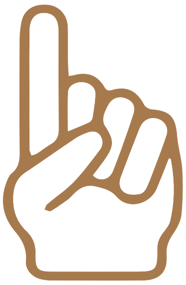
Open palm

The "Open palm" gesture is detected when all 5 fingers are stretched, resulting in the hand being "open".



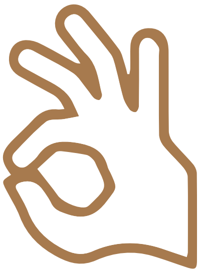
Point

The "Pointing" gesture is detected when the index finger is stretched and the other 4 fingers are folded. In case the thumb is stretched as well, this will still be recognised as a "Pointing" gesture.



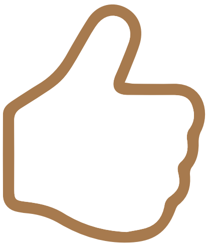
OK

The "OK" gesture is detected when the tip of the index finger and thumb are touching (resulting in a loop) and the other 3 fingers are stretched.



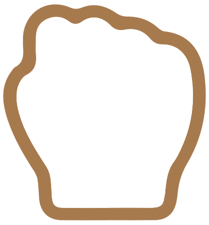
Thumb

The "Thumb" gesture is detected when the thumb is stretched and the other 4 fingers are folded.



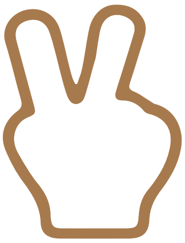
Fist

The "Fist" gesture is detected when all 5 fingers are completely folded.



V

The "V" gesture is detected when te index finger and middle finger are stretched and the other 3 fingers are folded.



The following gestures are dynamic gestures, meaning that they will be recognized when a sequence of hand movements are executed. As a result, dynamic gestures will typically have a higher non-detection rate in the field, as the sequence of hand movements can be executed differently, depending on the interpretation of the user.

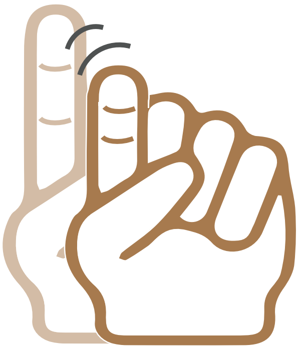
Swipe

A "Swipe" gesture is detected when a person's hand moves from left to right in a smooth motion. The following will make it easier for the sensor to recognize the swipe gesture:

- Raise the hand at torso level, not higher or lower.
- Keep the hand in front of the torso or arm, don't spread the arms to the side.
- Instead of making the movement with the entire arm, make the swipe movement from the wrist or elbow.
- Within your swiping movement, start with the palm of the hand (slightly) directed towards the camera, and end with the back of the hand (slightly) directed towards the camera. Or vice versa.
- Make sure the hand always remains completely in the Field of View of the sensor
- Perform the swipe gesture at a natural speed. A slower swipe will not be recognized easier than a faster movement. If any, a slightly swifter swipe is easier for the sensor to recognize.

Tap

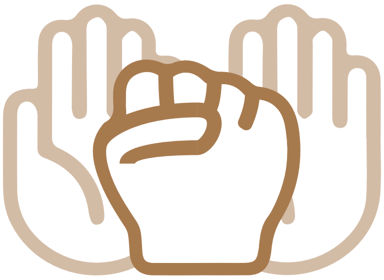
A "Tap" gesture is detected when a person's hand is in "Point" position (see previous page) and then swiftly moves the index finger to the front and back (as if tapping in the air). The following will make it easier for the sensor to recognize the swipe gesture:



- Make sure the palm of the hand is directed towards the camera. A tap gesture is not recognised from a side position.
- A tap gesture in the center of the sensor's Field of View is easier recognised then a tap gesture on the outer areas.

Grab

A "Grab" gesture is detected when a person's hand is in "Open Palm" position (see previous page) and then swiftly moves to "Fist" position and back to "Open Palm" position. The following will make it easier for the sensor to recognize the swipe gesture:



- Make sure movement sequence "Open palm - Fist - Open palm" is executed swiftly. When performed to slow, the gestures not be recognized as a "Grab", but instead as 3 separate gestures (Open Palm, Fist, Open Palm).

5 - Settings - Sensor behaviour

The XV HandGesture sensor has multiple settings which determine the behaviour of the sensor. The settings can be adjusted by sending X-talk setting commands via the API. After a power cycle, the settings always return to back to default.

Setting 1: Status LED behaviour

1. LED on	X001S[1:1]
2. LED off	X001S[1:2]
3. LED on, blink on trigger	X001S[1:3]
4. LED off, blink on trigger (default)	X001S[1:4]

Setting 4: Trigger on 1 or 2 hands*

1. Only trigger on closest hand	X001S[4:1]
2. Only trigger on hand most in centre	X001S[4:2]
4. Trigger on both hands (def)	X001S[4:4]

This setting determines if the sensor will trigger API messages for 2 detected hands, or just one. This setting applies for Gesture detection, Position tracking and Activation zones.

Setting 8: Filter level for gesture detection trigger*

Set filter Level	X001S[8:X]

X is a value between **1-20** and its default value is **2**. The filter level determines how many consecutive frames a gesture must be detected in order for it to trigger an API command. Especially when moving, people can briefly place their hand(s) in a position which the sensor detects as a valid hand gesture. This setting is intended to decrease/eliminate the amount of undesired ghost triggers. Increasing the filter level decreases the amount of ghost triggers, but it will also decrease the responsiveness of the sensor. Vice versa, decreasing X will increase the responsiveness of the sensor but can also result into more ghost triggers, as it will take less consecutive frames of a hand gesture being recognized for it to trigger an API command.

Setting 9: Minimum hand size for trigger*

Set minimum hand size	X001S[9:X]

X is a value between **1-100** and its default value is **10**. It indicates the minimum hand size (in % of the height of the field of view), for a detected hand to trigger an API message .

Setting 12: Interval for Position tracking triggers

1. Only output at Gesture detection	X001S[12:1]
2. Trigger every 250ms	X001S[12:2]
3. Trigger every 500ms	X001S[12:3]
4. Trigger every 1000ms	X001S[12:4]
5. No triggers, use data requests (d)	X001S[12:5]

This setting determines when an API message with Position tracking information is triggered. Please see page 6 for more information on Position tracking.

Setting 15: Activation zone triggers

1. Enable for any detected hand (def)	X001S[15:1]
2. Enable for any detected gesture	X001S[15:2]
3. Enable for any activated gesture	X001S[15:3]
4. Disable Activation zone triggers	X001S[15:4]

This setting determines when an API message with Activation zone information is triggered. Setting 15:3 refers to gestures which are activated via setting 20-28.

Setting 16: Reference point for Position tracking

1. Centre of hand	X001S[16:1]
2. Tip of Index finger (default)	X001S[16:2]
3. Tip of Middle finger	X001S[16:3]
4. Tip of Ring finger	X001S[16:4]
5. Tip of Pinky	X001S[16:5]
6. Tip of Thumb	X001S[16:6]

This setting determines the reference point for the X and Y coordinate information in the Position tracking API messages. This setting al so applies as the reference point for Activation zone entering and exiting. After adjusting this setting a START! command must be send. Please see page 11 for more information.

Setting 17: Hysteresis level for Activation Zones

Set hysteresis level	X001S[17:X]

X is a value between **1-5** and its default value is **2**. It indicates the hysteresis threshold (in % of the width of the field of view), for entering and exiting an Activation zone.

This setting prevents that consecutive enter/exit triggers are send when the reference point of a detected hand is on the edge of an Activation zone.

** These settings do not affect the AI engine, and thereby also don't affect the detected hands shown in the HDMI output. They only serve as a filter for API message triggers.*

5 - Settings - API output and Gesture triggers

The XV HandGesture sensor has multiple settings which determine the API output of the sensor. Next to this, API message triggers can be activated and deactivated for each gesture. The settings can be adjusted by sending X-talk setting commands via the API. After a power cycle, the settings always return to back to default.

Setting 5: Output mode for Gesture detection

1. Gesture only	X001S[5:1]
2. Gesture w/ direction (def)	X001S[5:2]
3. Gesture w/ Direction and F/B	X001S[5:3]
4. Gesture w/ Direction and ID	X001S[5:4]
5. Gesture w/ Direction and L/R	X001S[5:5]
6. Gesture w/ Direction, F/B and ID	X001S[5:6]
7. Gesture w/ Direction, F/B and L/R	X001S[5:7]
8. Gesture w/ Direction, ID and L/R	X001S[5:8]
9. Gesture w/ Direction, F/B, ID and L/R	X001S[5:9]
10. No triggers, use data requests	X001S[5:10]

For more information on the different output modes for gesture detection and the corresponding API messages, please see page 4.

Setting 6: Output format for Gesture detection

1. Parameters in 1 API message (d)	X001S[6:1]
2. Parameters in separate messages	X001S[6:2]

For more information on the output formats for gesture detection and the corresponding API messages, please see page 4.

Setting 11: Output mode for Position tracking

1. Output X, Y coordinate only	X001S[11:1]
2. Output X, Y coordinate and size (d)	X001S[11:2]
3. Output X, Y coordinate and angle	X001S[11:3]
4. Output X, Y, size and angle	X001S[11:4]

This setting determines what parameters are included in the API message when position tracking is enabled (see setting 12). For more information on the different output modes for position tracking and the corresponding API messages, please see page 6.

Setting 13: Output format for Position tracking

1. Parameters in 1 API message (d)	X001S[13:1]
2. Parameters in separate messages	X001S[13:2]

This setting determines the output format of the API message when position tracking is enabled (see setting 12). For more information on the output formats for gesture detection and the corresponding API messages, please see page 6.

Setting 20: Activate/deactivate trigger "Open palm" *

1. Activate "Open palm" trigger (def)	X001S[20:1]
2. Deactivate "Open palm" trigger	X001S[20:2]

Setting 21: Activate/deactivate trigger "Pointing" *

1. Activate "Pointing" trigger (def)	X001S[21:1]
2. Deactivate "Pointing" trigger	X001S[21:2]

Setting 22: Activate/deactivate trigger "Thumb" *

1. Activate "Thumb" trigger (def)	X001S[22:1]
2. Deactivate "Thumb" trigger	X001S[22:2]

Setting 23: Activate/deactivate trigger "OK" *

1. Activate "OK" trigger (def)	X001S[23:1]
2. Deactivate "OK" trigger	X001S[23:2]

Setting 24: Activate/deactivate trigger "Fist" *

1. Activate "Fist" trigger (def)	X001S[24:1]
2. Deactivate "Fist" trigger	X001S[24:2]

Setting 25: Activate/deactivate trigger "V" *

1. Activate "V" trigger (def)	X001S[25:1]
2. Deactivate "V" trigger	X001S[25:2]

Setting 26: Activate/deactivate trigger "Swipe" *

1. Activate "Swipe" trigger (def)	X001S[26:1]
2. Deactivate "Swipe" trigger	X001S[26:2]

Setting 27: Activate/deactivate trigger "Tap" *

1. Activate "Tap" trigger (def)	X001S[27:1]
2. Deactivate "Tap" trigger	X001S[27:2]

Setting 28: Activate/deactivate trigger "V" *

1. Activate "Grab" trigger (def)	X001S[28:1]
2. Deactivate "Grab" trigger	X001S[28:2]

** These settings do not affect the AI engine, and thereby also don't affect the detected hands shown in the HDMI output. They only serve as a filter for API message triggers.*

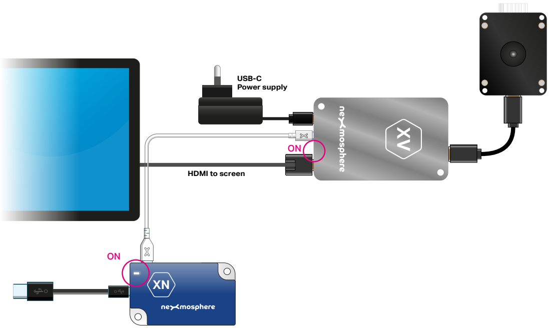
6. Quick test

In order to test if the XV HandGesture sensor is installed correctly, please follow the test procedure below:

Step 1 - Setup

First, connect the XV sensor to an Xperience controller and connect a screen to the HDMI output of the XV sensor. Then, power the sensor and secondly the Xperience controller as well.

The white status LED of the XV sensor should go on. The status LED of the controller will start to blink and once power-up is completed will be lit continuously. It will take approximately 60 seconds for the sensor get into operational mode.



Step 2 - Test gesture detection on screen

When the sensor is operational, a live camera feed will appear on he screen. When ready, make a thumbs-up gesture in front of the camera.

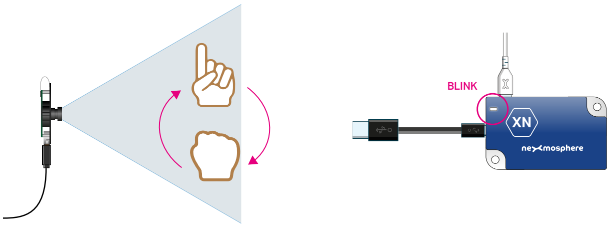
A text will appear on the screen above the detected hand, indicating that a "thumb up" gesture was recognized.

In case the video feed is blurry, rotate the lens until the video feed is in focus on the hand making the gesture.

Step 3 - Test API triggers

Make a "pointing" gesture, then make a "fist" gesture. Keep alternating between them with a few seconds in between,

Each time a new gesture is recognized, the status LED of the controller should blink once.



In case any of the 3 steps above does not provide the expected result, please check the installation guidelines in this document.

For a full test we recommend to connect the setup to a mediaplayer or PC and test all API commands listed in this document (see section 3). For more information on how to setup a test for your controller, please see the Quick Start Guide of the Xperience controller you are using. These are available on nexmosphere.com/support-documentation

Please contact support@nexmosphere.com for any support questions you may have.