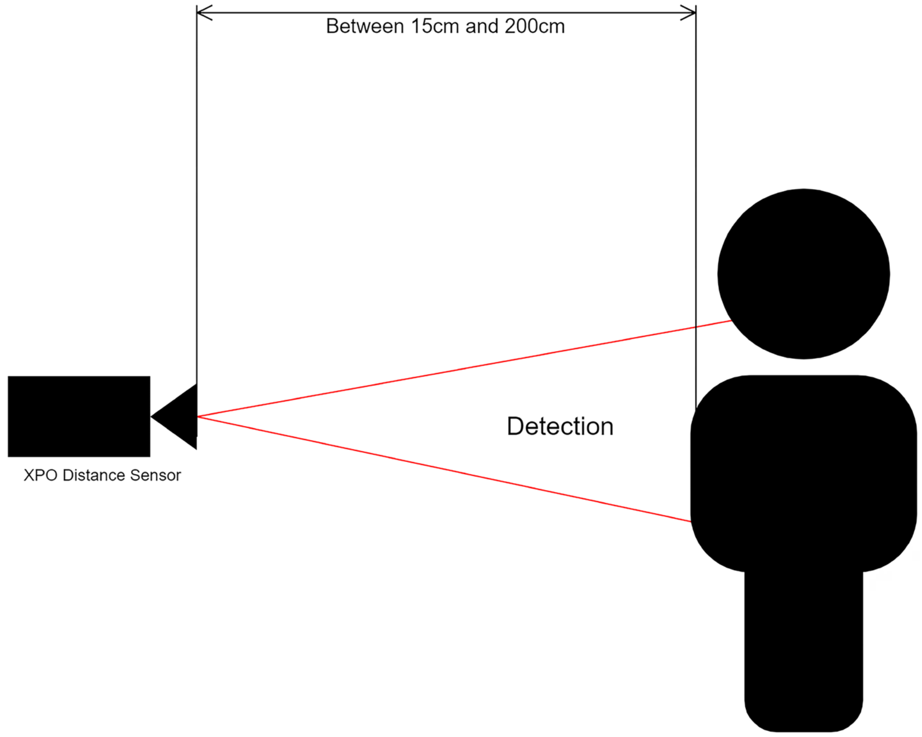
Third party sensor integration

Brightsign displayboard



This document describes the integration that can be made with a XPO Screens' displayboard + sensorboard combination in order to get communication working, receiving and sending commands as well as calibrating said sensors.

Table of Contents

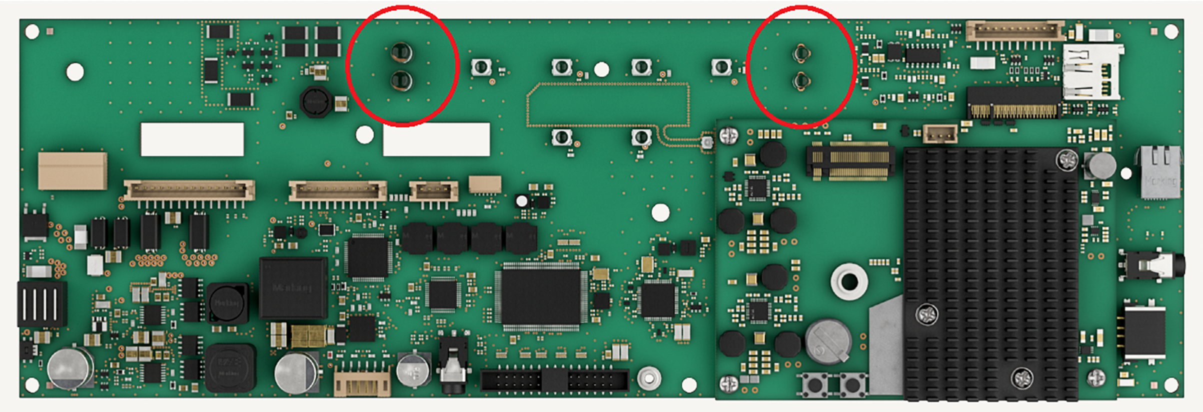
Command list

To download the full command list scan the QR code or click here.

Quick overview

Our displayboards have been kitted out with distance sensors in two variations:

1. On-board with our BrightSign embedded series displayboard:



2. As a separate module for our HDMI displayboard or additional sensorboard for the BrightSign embedded displayboard:

Both methods have the same communication protocol, a serial connection which uses readable commands for all keywords and parameters. There is a comprehensive list of all commands in the last section of this document.

We've also made an OS Extension specifically for BrightSign mediaplayers that further simplify the process of sensor communication. For more information on the XPO Core, our BrightSign OS Extension see this document.

Serial connection setup & config

The serial connection can be set-up and tested using any Windows or Linux-based machine. Quick tests can even be made with a mobile phone and a OTG adapter.

Below are some general settings for the serial port:

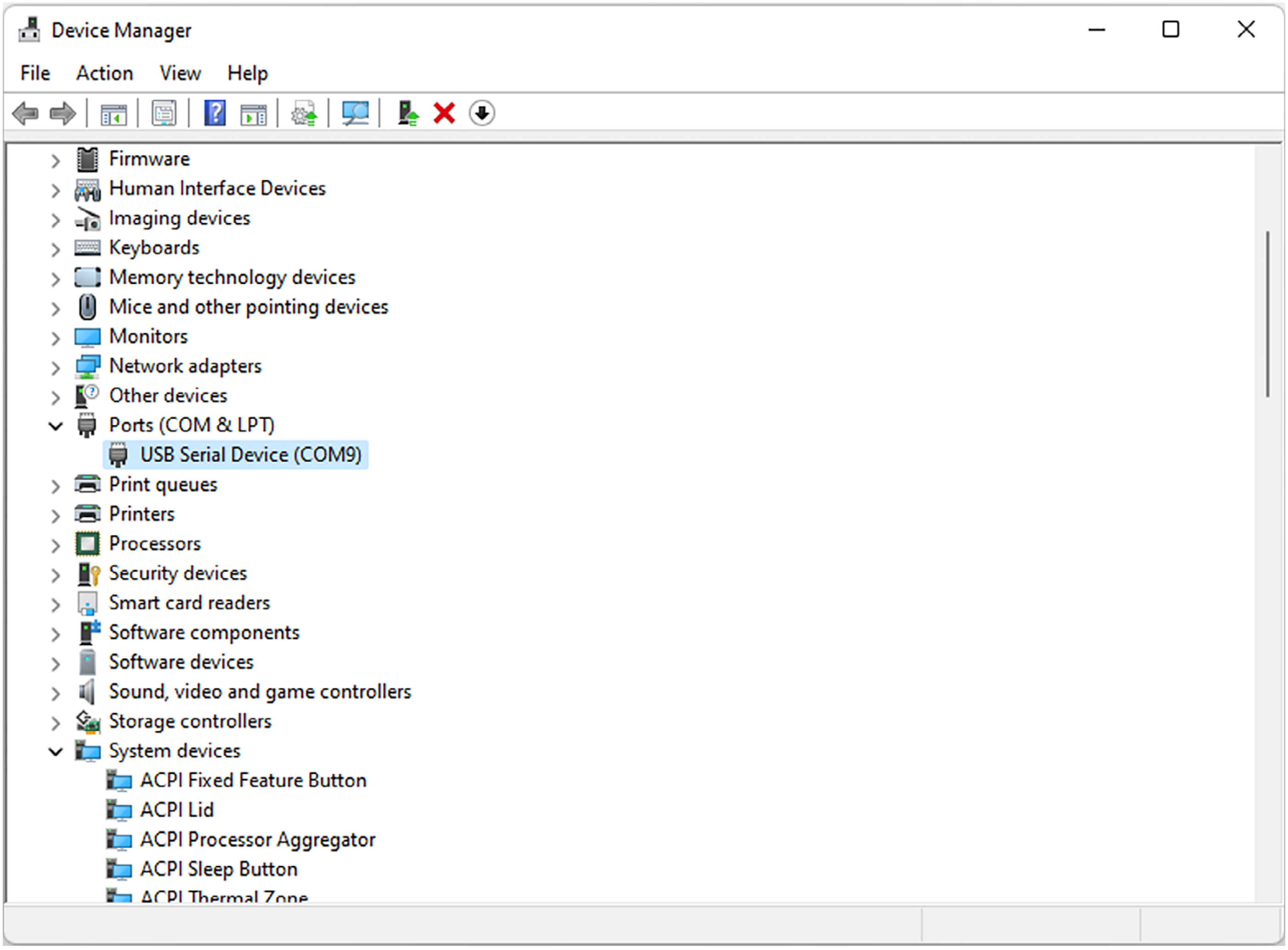
Name	Value
Protocol	ASCII
Baud rate	115200
Data bits	8
Parity	None
Stop bits	1
Send EOL	CR
Receive EOL	CR

Hardware setup

The sensorboard cannot be attached to a device directly, it always has to go through a XPO Screens' displayboard, the displayboard can then be connected to a computer or mediaplayer using a USB cable.

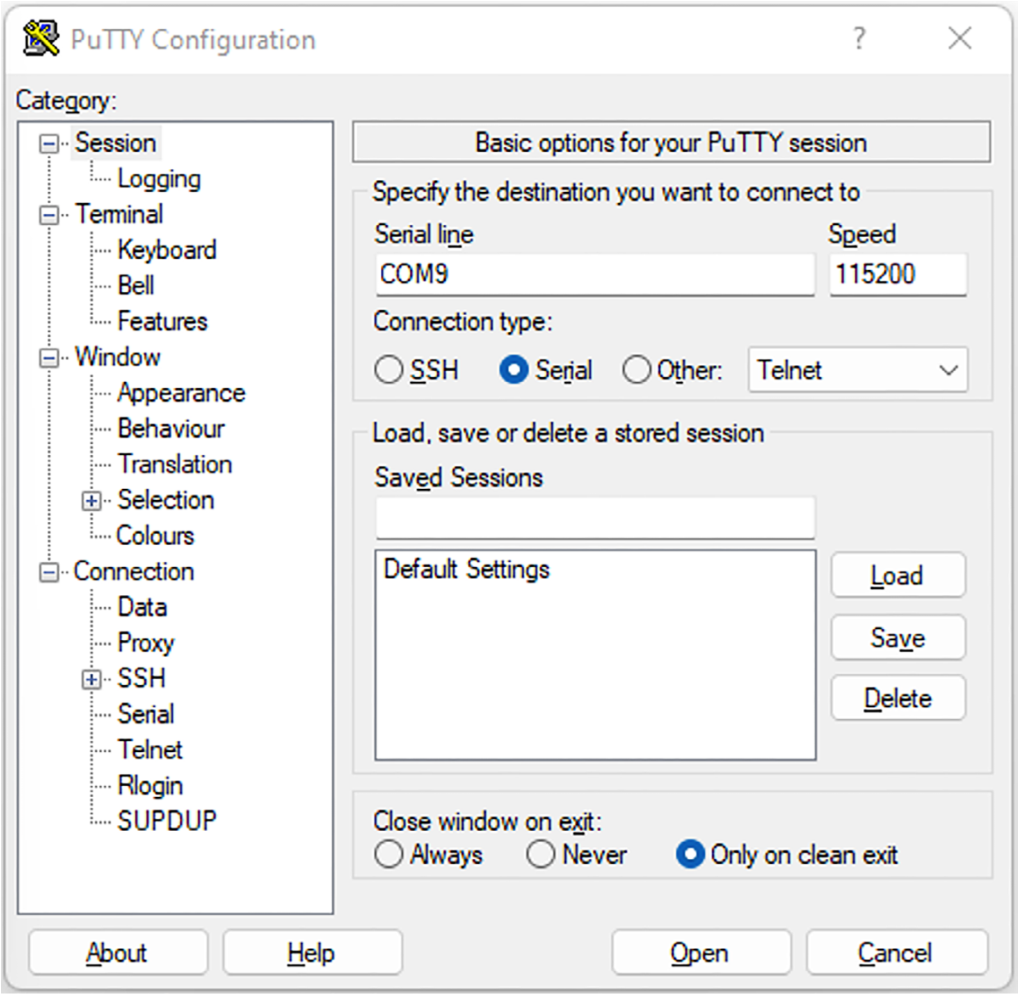
Software setup

When a displayboard is plugged in the computer we need to know which COM port it is assigned to, to see this open "device manager" on Windows or use "hardinfo" on Linux.



- Firmware
- Human Interface Devices
- Imaging devices
- Keyboards
- Memory technology devices
- Mice and other pointing devices
- Monitors
- Network adapters
- Other devices
- Ports (COM & LPT)
  - USB Serial Device (COM9)
- Print queues
- Printers
- Processors
- Security devices
- Smart card readers
- Software components
- Software devices
- Sound, video and game controllers
- Storage controllers
- System devices
  - ACPI Fixed Feature Button
  - ACPI Lid
  - ACPI Processor Aggregator
  - ACPI Sleep Button
  - ACPI Thermal Zone

When the correct port is found we can open a program like PuTTY (for Windows, Linux & Mac) to test our serial communication. We insert the COM port and set the Baud rate of 115200 as our speed.



PuTTY Configuration

Category:

Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: COM9

Speed: 115200

Connection type: ○ SSH ● Serial ○ Other: Telnet

Load, save or delete a stored session

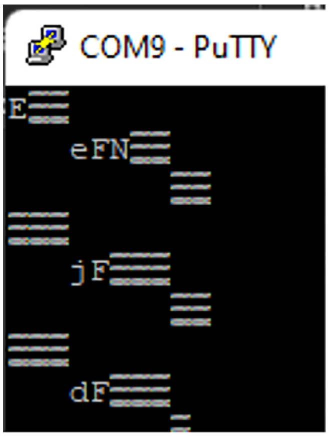
Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Upon opening the port we'll see one of two things:

1. Instantly see text-based commands being pushed, in this case skip the next step.
2. See either a jitter of text, unknown characters or a large combination of digits like this:



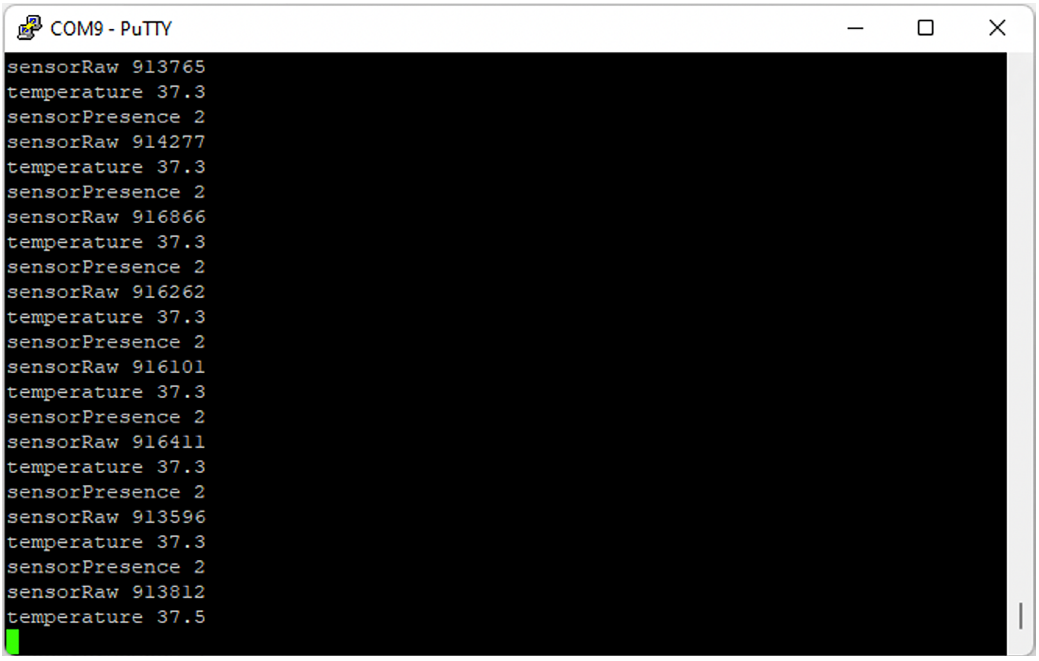
COM9 - PuTTY

```
E
    eFN

 
    jF

 
    dF
```

In this case type the following command: "enableTextMode" or "getSerialNumber" and hit enter. Any command can be executed like this from PuTTY, we should see an immediate change in the output.

In some programs there is a separate text input field for commands, in PuTTY you can simply start typing as soon as the connection is initiated. Hit "enter" to confirm a command.

Afterwards it should look something like this:



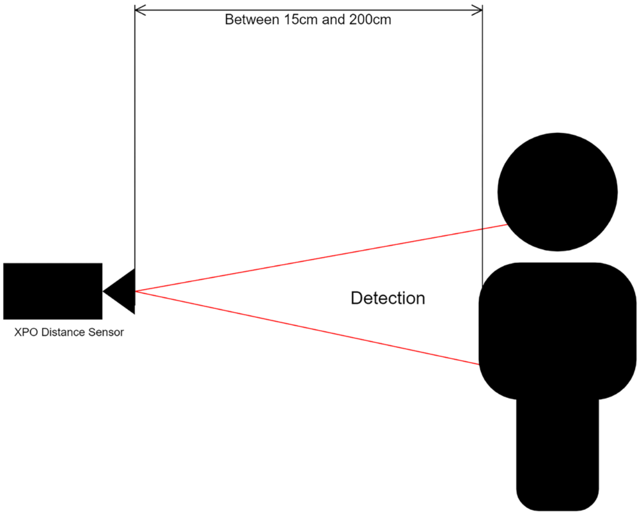
```
sensorRaw 913765
temperature 37.3
sensorPresence 2
sensorRaw 914277
temperature 37.3
sensorPresence 2
sensorRaw 916866
temperature 37.3
sensorPresence 2
sensorRaw 916262
temperature 37.3
sensorPresence 2
sensorRaw 916101
temperature 37.3
sensorPresence 2
sensorRaw 916411
temperature 37.3
sensorPresence 2
sensorRaw 913596
temperature 37.3
sensorPresence 2
sensorRaw 913812
temperature 37.5
```

If nothing shows on the screen regardless insert the following command: "setOOBFields sensorRaw sensorPresence temperature" and hit "enter".

This makes sure you should be able to see all available sensordata coming from the displayboard.

Distance sensor hardware introduction

The distance sensor is a custom infrared-based sensor that has its own computing onboard to calculate the signal and return useful values to the user. The sensors are directed perpendicular to the PCB they are on and have a working range of between 15cm and 200cm. Meaning that any closer than 15cm the sensor start to drop instead of increase and any further then 200cm means it stops showing meaningful deviations in the sensorvalues. The sensors have a working area that is slightly cone shaped, around 5 degrees, so anyone that passes by can be detected more accurately.



The sensors come with a black masking foam that makes sure they only reflect light to the front instead of sending and receiving directly from and to the sensor. Make sure it is applied correctly.

Distance sensor functionality introduction

When initiating a connection, we see three different data objects coming from the displayboard, sensor presence, sensor raw & temperature. The sensor presence & sensor raw come from the sensorboard, the temperature is the current temperature in degrees Celsius of the video converter chip.

The sensor raw is the raw value coming from the sensor, it is not a measurement of distance but a measurement of signal strength. The fallout of the sensor is logarithmic, meaning the closer you place an object near the sensor the faster the value starts to increase.

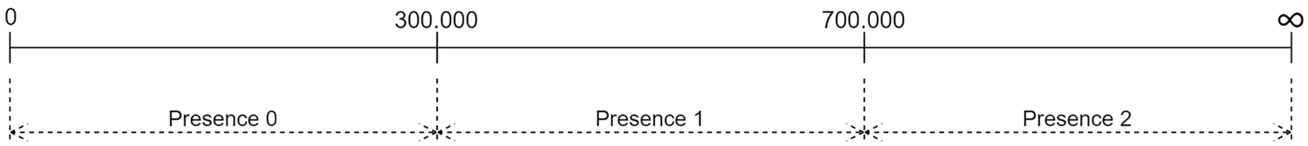
The sensors are usually fitted behind glass, but the XPO Screens sensors can also be fitted behind one-way mirror foil. This is done with a different mode which can be enabled using the serial communication (see command: setSensorSetting).

The sensor presence is always one of the following numbers: 0, 1 or 2.

These numbers are regions that can be defined by setting thresholds. 0 means no user is detected, 1 usually means a user has been detected and is currently a bit further away, 2 can be set to detect persons very close to the sensor. NOTE; this doesn't mean we have to use presence 2 at all. The thresholds can be setup to use only presence 0 for no detection and presence 1 for any detection. Only if there is need for 2 levels of detection will we have to calibrate the thresholds for detection zone 2.

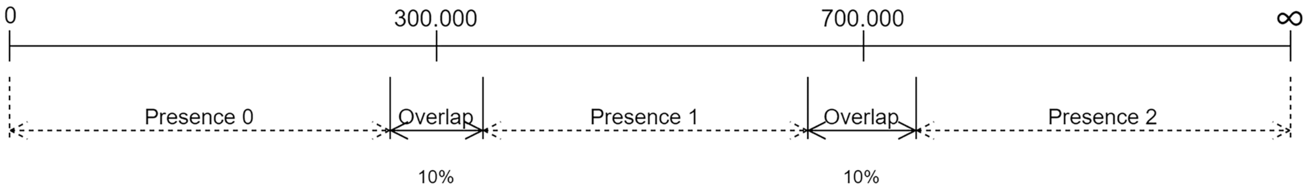
All thresholds in the following diagrams are samples, any actual sensor thresholds can be found using the sensor-raw value, more on that in the next part.

In the diagram below we see the different ranges of raw sensorvalues which correspond to a presence value, these raw values can then be used to determine the sensor thresholds.

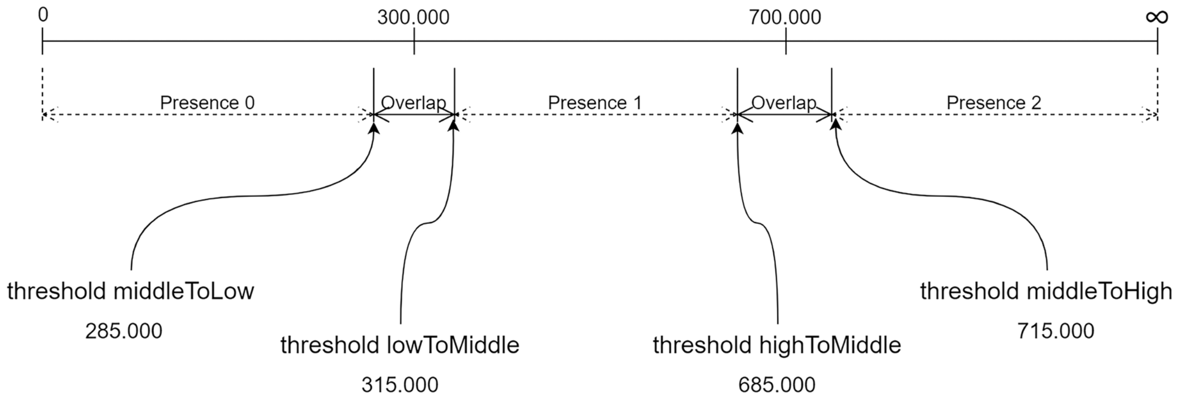


Any calibration requires the user to be able to read the raw sensorvalues, so before calibrating make sure we have an active connection and see the sensorvalues change every half second on the serial readout device used.

In the diagram below we see the addition of the overlap, this is a "buffer" zone to make sure our sensor is not constantly switching between presence 0 and presence 1 when the raw value happens to fluctuate around the switching point. There should be around 10% of buffer zone around the switching point. For a value of around 300.000 this means 30.000, so 15.000 below and 15.000 above.



In the diagram below we see how these values would correspond to the actual thresholds. So if we determine 300.000 is a good level we would take 10% around the value as a buffer zone. Creating 2 new values, 285.000 and 315.000. These will become the thresholds middleToLow and lowToMiddle. What that means is in order for the presence to go from 0 (low) to 1 (middle) the raw sensorvalue MUST exceed the value defined in lowToMiddle (being 315.000 in this example) and for the presence to go from 1 to 0 the raw sensorvalue MUST subceed the value defined in middleToLow (being 285.000 in this example).



When calibrating (full process explained in the next part) a good rule of thumb in defining the thresholds is to stand where we would like the sensors to "detect" us. Take a small step forward (lowToMiddle) and a small step backwards (middleToLow). That gives us pretty accurate values keeping around a 10% margin.

Distance sensor calibration 2 levels (simple detection)

The calibration process for the XPO Distance Sensor using presence 0 & 1 is a sequence consisting of:

- Setting up serial connection
- Reading raw sensorvalues
- Having a person stand in front of the sensors at the desired distance
- Making a note of the raw sensorvalue (as lowToMiddle)
- Taking 1 step back or around 30cm (increasing the distance between the sensor and the user)
- Making a second note of the raw sensorvalue (as middleToLow)
- Using the values noted down and sending them in command form to the displayboard over the serial connection

The exact commands to set the threshold are:

- setSensorThreshold middleToLow 285000
- setSensorThreshold lowToMiddle 315000

To check the current levels of thresholds use:

- getSensorThresholds

If the calibration went correctly we can see the presence change from 0 to 1 when the raw sensorvalue exceeds 315.000 and change back from 1 to 0 when the value subceeds 285.000. If calibration failed please try to determine the thresholds again or check the troubleshooting below.

Distance sensor calibration 3 levels (detection for far and near)

The calibration process for the XPO Distance Sensor using presence 0, 1 & 2 is a sequence consisting of:

- Setting up serial connection
- Reading raw sensorvalues
- Having a person stand in front of the sensors at the desired distance for far user detection
- Making a note of the raw sensorvalue (as lowToMiddle)
- Taking 1 step back or around 30cm (increasing the distance between the sensor and the user)
- Making a second note of the raw sensorvalue (as middleToLow)
- Having the person stand close to the sensor at the desired distance for near user detection
- Making a third note of the raw sensorvalue (as middleToHigh)
- Taking 1 step back or around 30cm (increasing the distance between the sensor and the user)
- Making a fourth note of the raw sensorvalue (as highToMiddle)
- Using the values noted down and sending them in command form to the displayboard over the serial connection

The exact commands to set the threshold are:

- setSensorThreshold middleToLow 285000
- setSensorThreshold lowToMiddle 315000
- setSensorThreshold highToMiddle 685000
- setSensorThreshold middleToHigh 715000

To check the current levels of thresholds use:

- getSensorThresholds

If the calibration went correctly we can see the presence change from 0 to 1 when the raw sensorvalue exceeds 315.000 and change back from 1 to 0 when the value subceeds 285.000. As well as see the presence change from 1 to 2 when the raw sensorvalue exceeds 715.000 and change back from 2 to 1 when the value subceeds 685.000. If calibration failed please try to determine the thresholds again or check the troubleshooting below.

Backlight controls

The sensors can be tied to some functions controlling the backlight, first we check some basic backlight controls.

Backlight level

Using the following command we can set the backlight intensity to a specific percentage, backlight levels can be set to any value between 0 and 100. With 100% being the brightest the panel backlight can be, and 0% being disabled entirely (caution should be used when setting the backlight to 0% because the user won't see anything appearing on screen anymore to aid them in turning it back on).

The command is: "setBacklightLevel 70" but can be set to anything between 0 and 100 although going lower than 30 is never recommended.

Backlight quick disable

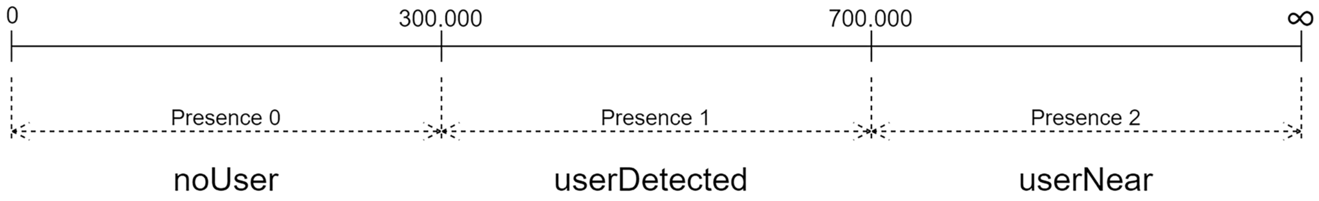
The backlight can be toggled on and off quickly using the following command: "setBacklightDisabledOverride true" will disable the backlight immediately, this value is independent of the backlight level, that will remain the same. The backlight will be enabled directly using the same command but setting the value to false.

Backlight off delay (sensor dependent)

The backlight can be toggled on/off automatically using the sensorpresence value described in the parts above. There are three commands involved: "setBacklightOnDetectionEnabled, setBacklightZones & setBacklightDelay".

For the mechanism to start working we first set the command "setBacklightOnDetectionEnabled true".

Second we define the zones where we want the backlight to toggle, choose a selection from the diagram below, note the actual zone names (i.e. noUser, userDetected, userNear)



So if we would like the backlight to turn on when anyone comes anywhere near our sensors we select both userDetected and userNear, this then goes into the command:

"setBacklightZones userDetected userNear" any parameter is just added, the sequence does not matter, use spaces between the parameters. What this specific command will now do is make sure the backlight turns on whenever the userPresence becomes 1 OR 2. And will turn off when the presence becomes 0.

The third command, setBacklightDelay determines how long the backlight will remain on after the last presence was detected, the parameter is a number between 0 and 3600 defined in seconds. For this mechanism to work a minimum of 1 second must be set. So "setBacklightDelay 1800" will make sure that even though the sensorpresence becomes 0, the backlight will remain on for half an hour. This enhances the experience while at the same time conserve electricity and durability of the hardware.

LED settings

The LEDs on the sensorboard can be changed to any value with the variant on the BrightSign embedded displayboard having fully addressable RGBW LEDs. The external sensorboard only has white LEDs and the HDMI displayboard is incompatible with the RGBW LEDs of the BrightSign embedded displayboard.

For the white only LEDs the light level is set using the following command:

"setLED all 150 default", where 150 is the intensity value between 0 and 255.

For the RGBW LEDs the levels are set using the following command:

"setLED all FF0000 default", where FF0000 is the HEX colour code for RED in this case.

These LEDs can also be individually addressed using:

"setLED 4 00FF00 default", where 4 is the 5th LED (counting from 0 to 5 using 6 LEDs in total)

To use the white LED in change the HEX code to a value between 0 and 255 like this:

"setLED 4 210 default"

The keyword "default" is always used EXCEPT when the value should not be remembered when power cycling the board.

Troubleshooting

When there is mirror foil on the glass in front of the sensors:

Make sure to set the sensors to mirror mode using the following command:

"setSensorSetting mirror"

This increases the sensors' signal and lets it see through the mirror foil.

To reset the setting, send:

"setSensorSetting glass"

There are no values or only very low values:

Check all cables if they are inserted correctly, turn off the power and replug all connectors if necessary, mainly check the USB connector. Also make sure the sensors are actually connected,

Raw sensor values remain at 999.999.999

The sensors have been disabled, to enable them use the command:

"setSensorEnabled true"

The sensorpresence remains 0, even when standing in front of the sensors

Make sure nothing is obstructing the view of the sensors, your thresholds might not be set correctly, if they are too high and the raw sensorvalue never exceeds the lowest value the presence will never reach 1.

The sensorpresence remains 1, even when moving away from the sensors

Make sure nothing is obstructing the view of the sensors, your thresholds might not be set correctly, if they are too low and the raw sensorvalue never subceeds the highest value the presence will never reach 0.