



TOPOSENS

# Toposens TS3 Kit & Evaluation Set Advanced Ultrasonic Sensor Arduino Area Demo

## Demo Area Version

Revision	Date	Changes
V 1.0	08/2020	Initial Release

## Introduction

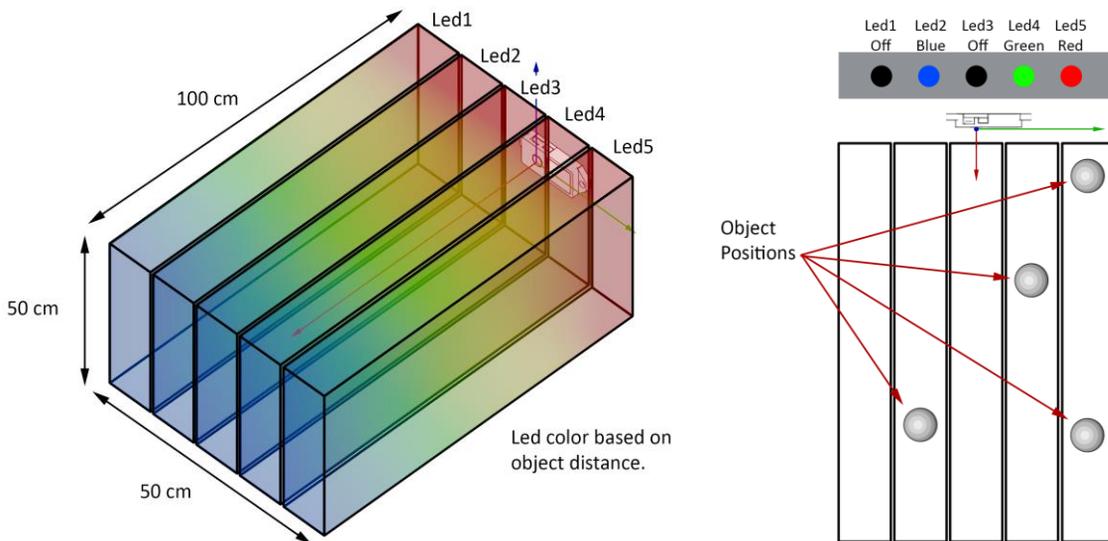
The Area Demo demonstrates the usage of the TS3 sensor in an exemplary use case. The code and parts to replicate this demo are available in the documentation folder of this document. The focus of this demo is the filtering technique to show how possible customer applications can use the TS3 data. The other functions are standard support functions which are not described in this documentation.



The demo is not a product. The code can be utilised for own projects. Toposens GmbH is not affiliated with Arduino AG nor does Toposens provide support regarding issues with Arduino products.

## Demo Functionality

The Arduino program shows the functionality of the TS3 as area surveillance tool. The demo code can be configured to fit different environments and use cases. The standard configuration is a small surveillance area as shown in the lower image. If an object is detected inside one of the cubes the according Led will light up in the color of the distance. If multiple objects are detected inside a cube, the closest object defines the Led color.

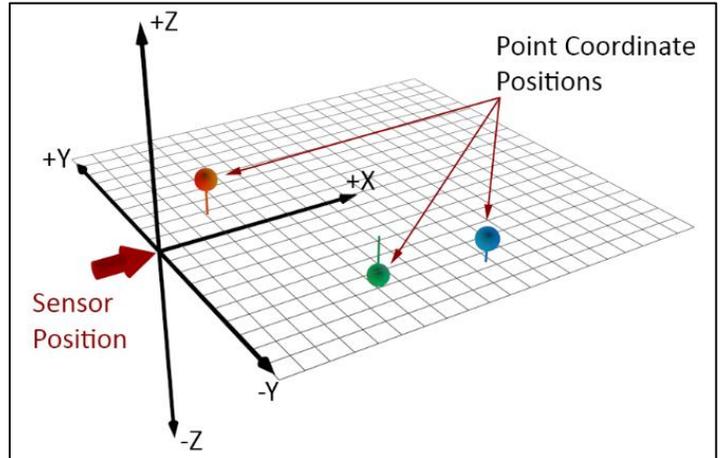


The configuration of the detection area can be changed in the Arduino Code. All functions except the Led-Color are parametric. The core technology of the area surveillance is a voxel filter which enables a low-level confidence level for each detection.

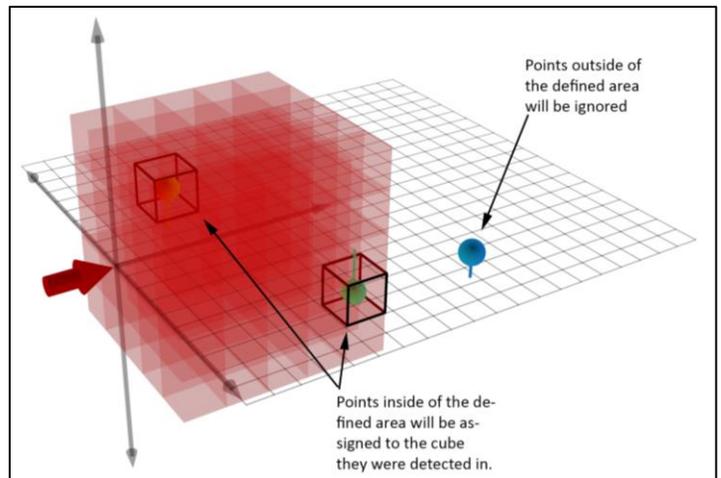
## Voxel Filter Principle

The TS3 sensor data is filtered by a filter which divides the area of interest into equally sized cubes. The cube size depends on the configured area and the division factors. Each spatial dimension can be configured separately, which enables more flexibility.

The TS3 raw data is parsed by the Arduino library. The following example shows 3 points at different positions in front of the sensor. The color of the points indicates the distance from the sensor.

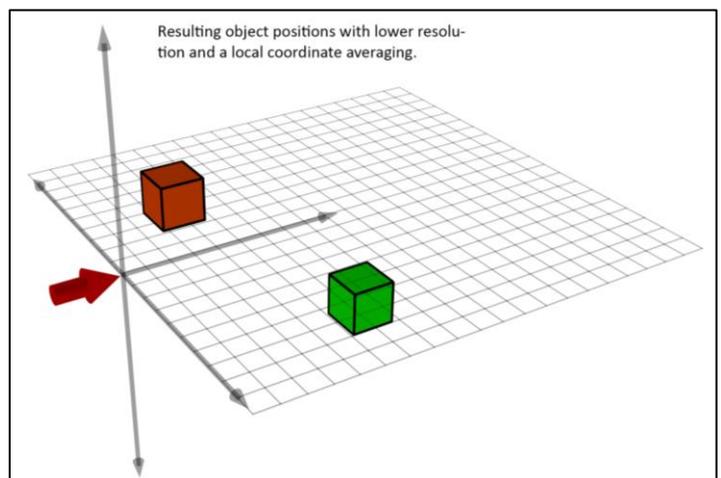


The spatial extension and resolution (=division into discrete sections or cubes) of the voxel array can be freely configured. If a point is detected within the defined array, the corresponding storage "bin" is incremented by the value of the volume of the point. The volume value can be divided to avoid instantly filled bins when loud objects are detected.



The resulting object coordinates are of lower resolution. Due to the averaging characteristics of the detections, the points are more stable. A second threshold based on the value stored in the array can be defined to only output array positions with a certain value in them, with the conclusion that either a significant echo was detected, or a less powerful echo for multiple frames.

Each voxel cell is decremented after each frame to avoid cluttering with irrelevant data.



# Voxel Filter Flowchart

