

**This document provides guidance on following topics:**

- 1. Description of functions available in the template.**
- 2. How to use the template file while writing an OpenSCAD script.**
- 3. Example use cases of functions illustrating their outcome.**

## **1. Description of functions available in the template.**

### **Function #1**

- **Syntax:** `close_path (points_set)`
- **Purpose:** Returns a new point set with the first point appended to the end.
- **Input:**
  - `points_set`: A list of 2D points `[[x, y], [x, y], ...]`

### **Function #2**

- **Syntax:** `show_points (points_set, radius, fragments, col)`
- **Purpose:** Displays each point as a sphere.
- **Input:**
  - `points_set`: A list of 2D points.
  - `radius`: Radius of the sphere at each point.
  - `fragments`: Controls smoothness of the sphere (number of fragments).
  - `col`: Color of the spheres (string name or RGB list).

### **Function #3**

- **Syntax:** create\_polygon (points\_set)
- **Purpose:** Draws a filled 2D polygon using the provided point set.
- **Input:**
  - points\_set: A list of 2D points in the order of polygon vertices.

### **Function #4**

- **Syntax:** extrude\_polygon (points\_set, height)
- **Purpose:** Extrudes a 2D polygon.
- **Inputs:**
  - points\_set: A list of 2D points in the order of polygon vertices.
  - height: Extrusion height.

### **Function #5**

- **Syntax:** create\_2d\_path (points\_set, width, fragments, shape)
- **Purpose:** Creates a 2D path using hulls between either circles or squares.
- **Inputs:**
  - points\_set: A list of 2D points.
  - width: Diameter (for circle) or side length (for square).
  - fragments: Controls smoothness (used only if shape is "circle").
  - shape: Shape type for drawing hull — "circle" or "square".

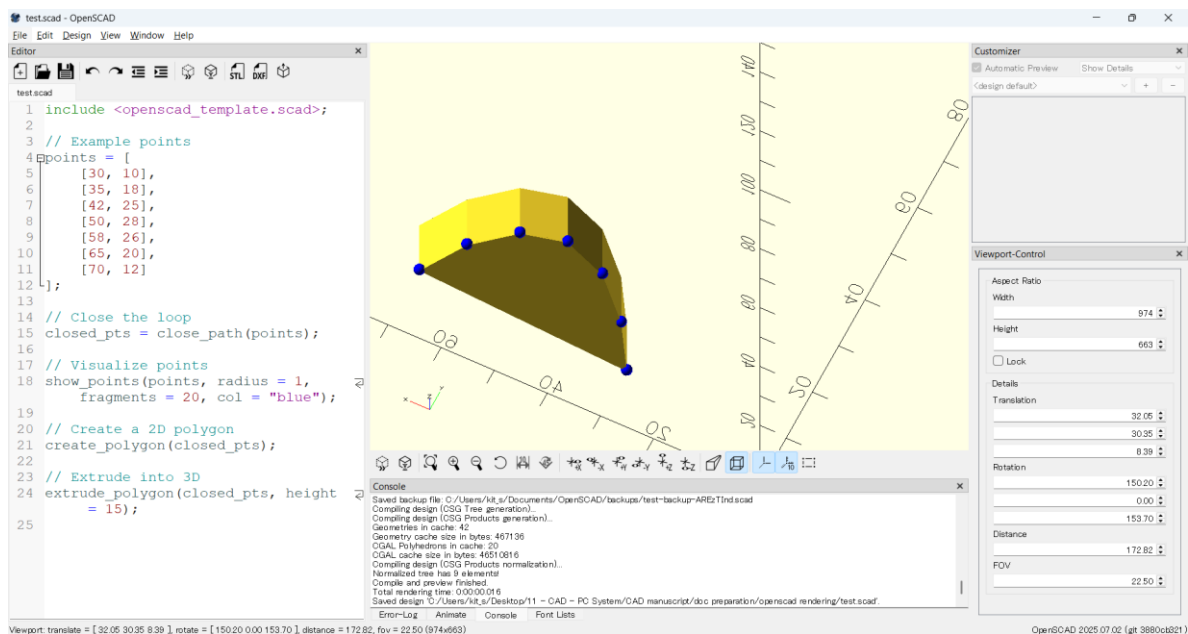
### **Function #6**

- **Syntax:** extrude\_2d\_path (points\_set, width, fragments, height, shape)
- **Purpose:** Extrudes a 2D path based on either circles or squares.
- **Inputs:**
  - points\_set: A list of 2D points.
  - width: Diameter (for circle) or side length (for square).
  - fragments: Controls smoothness (used only if shape is "circle").
  - height: Extrusion height.
  - shape: Shape type for drawing hull — "circle" or "square".

## 2. How to use the template file while writing an OpenSCAD script.

- (a) Download the template file (openscad\_template.scad) and place it in the same folder as your OpenSCAD script.
- (b) Include the template at the top of your script so the functions become available.
- (c) Define or paste your point set, typically obtained from Systems A–E.
- (d) Call and use the above functions to render geometry directly from the points.

An example code excerpt and related outcome are shown below:



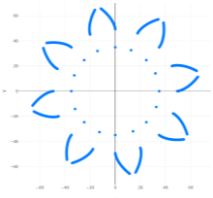
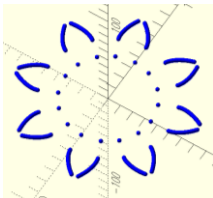
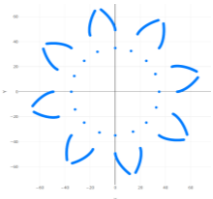

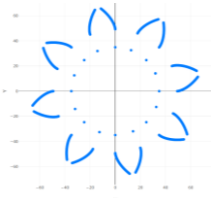
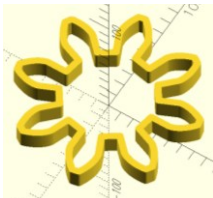
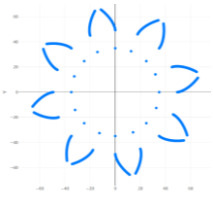
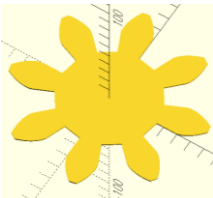
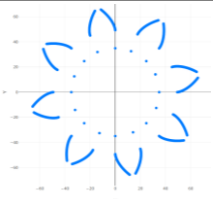

Note: We recommend using the OpenSCAD Development Snapshot builds for full compatibility with the template. All examples in this guide were created and tested using the OpenSCAD Development Snapshot for Windows (2025.07.02).

### 3. Example use cases of functions illustrating their outcome.

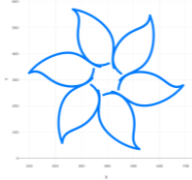

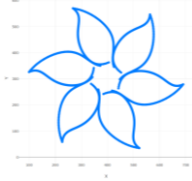

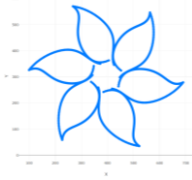




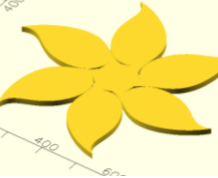
#A

Function(s) Used	Input Points	Outcome
<code>show_points (points_set = points, radius = 1.5, fragments = 70, col = "blue");</code>		
<code>create_2d_path (points_set=points, width=5, fragments=75, shape="circle");</code>		
<code>extrude_2d_path (points_set=points, width=5, fragments=75, height=15, shape="circle");</code>		
<code>create_2d_path (points_set=close_path (points), width=5, fragments=75, shape="circle");</code>		
<code>extrude_2d_path (points_set=close_path(points), width=5, fragments=75, height=15, shape="circle");</code>		
<code>create_polygon (close_path(points));</code>		
<code>extrude_polygon (close_path(points), height=15);</code>		

#B

Function(s) Used	Input Points	Outcome
<code>show_points (points_set = points, radius = 1.5, fragments = 70, col = "blue");</code>		
<code>create_2d_path (points_set=points, width=5, fragments=75, shape="circle");</code>		
<code>extrude_2d_path (points_set=points, width=5, fragments=75, height=15, shape="circle");</code>		
<code>create_polygon (points);</code>		
<code>extrude_polygon (points, height=15);</code>		

#C

Function(s) Used	Input Points	Outcome
<code>show_points (points_set = points, radius = 1.5, fragments = 70, col = "blue");</code>		
<code>create_2d_path (points_set=points, width=5, fragments=75, shape="circle");</code>		
<code>extrude_2d_path (points_set=points, width=5, fragments=75, height=15, shape="circle");</code>		
<code>create_polygon (points);</code>		
<code>extrude_polygon (points, height=15);</code>		

**Note:** Across the example use cases shown in this guide, it becomes evident that the developed functions allow point-cloud data to be converted into meaningful geometric forms with only a few lines of OpenSCAD code. Operations such as closing a boundary, constructing paths, forming polygons, or extruding shapes can be achieved directly from the point sets without requiring users to manually write complex modeling routines. The example datasets used throughout this documentation are the same point sets generated in the “How to Use System A–E” guides. Users who wish to understand how these point sets were created are encouraged to refer to those system-specific guides. This consistent reuse of datasets ensures continuity and demonstrates how point clouds produced in the IPCM Layer translate seamlessly into the Script-Based Rendering Layer.