

# SIRIUS2

v.1.25

©SPIRALLAB

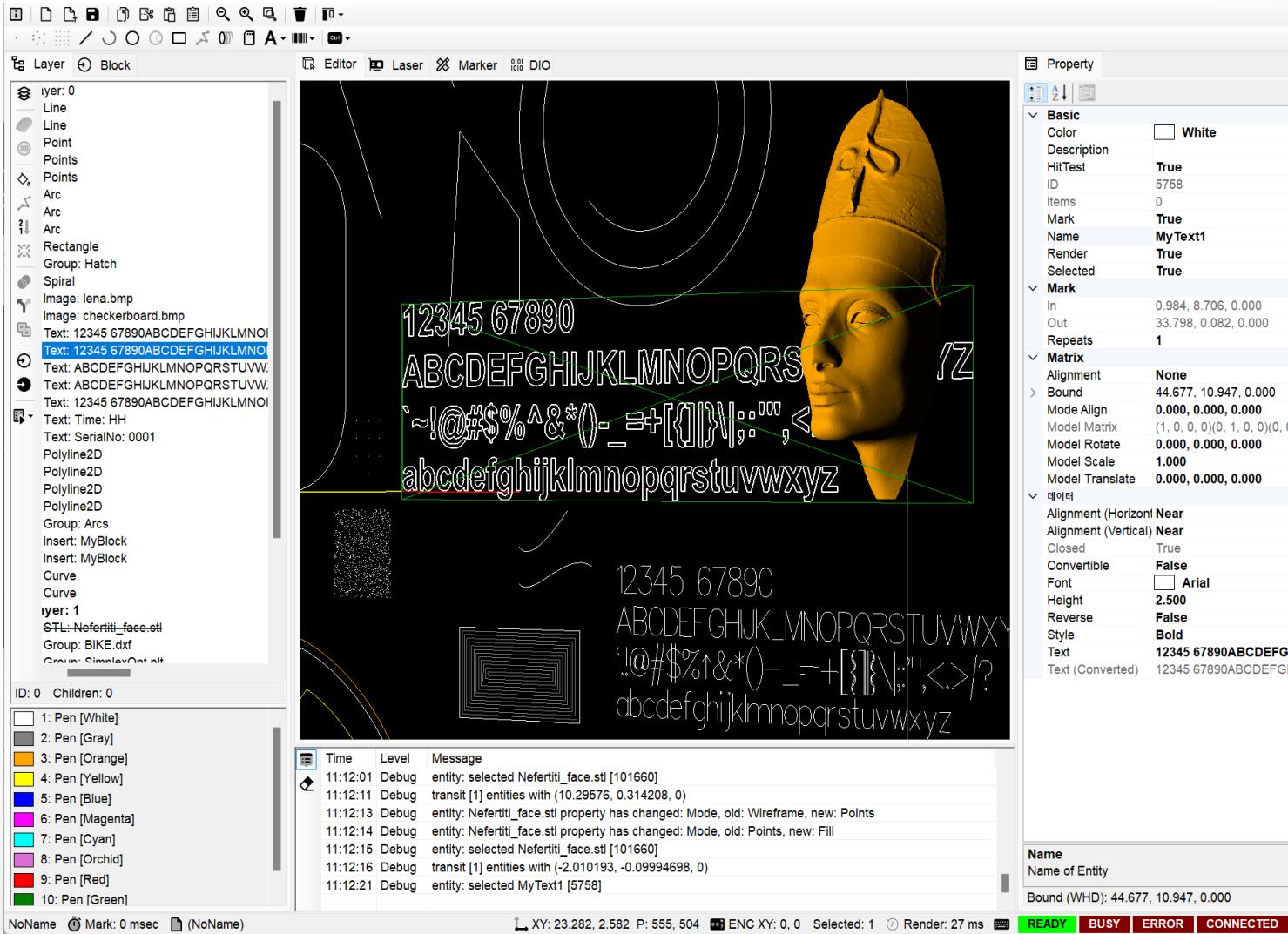
[hcchoi@spirallab.co.kr](mailto:hcchoi@spirallab.co.kr)

<http://spirallab.co.kr>

# What's Advantages ?

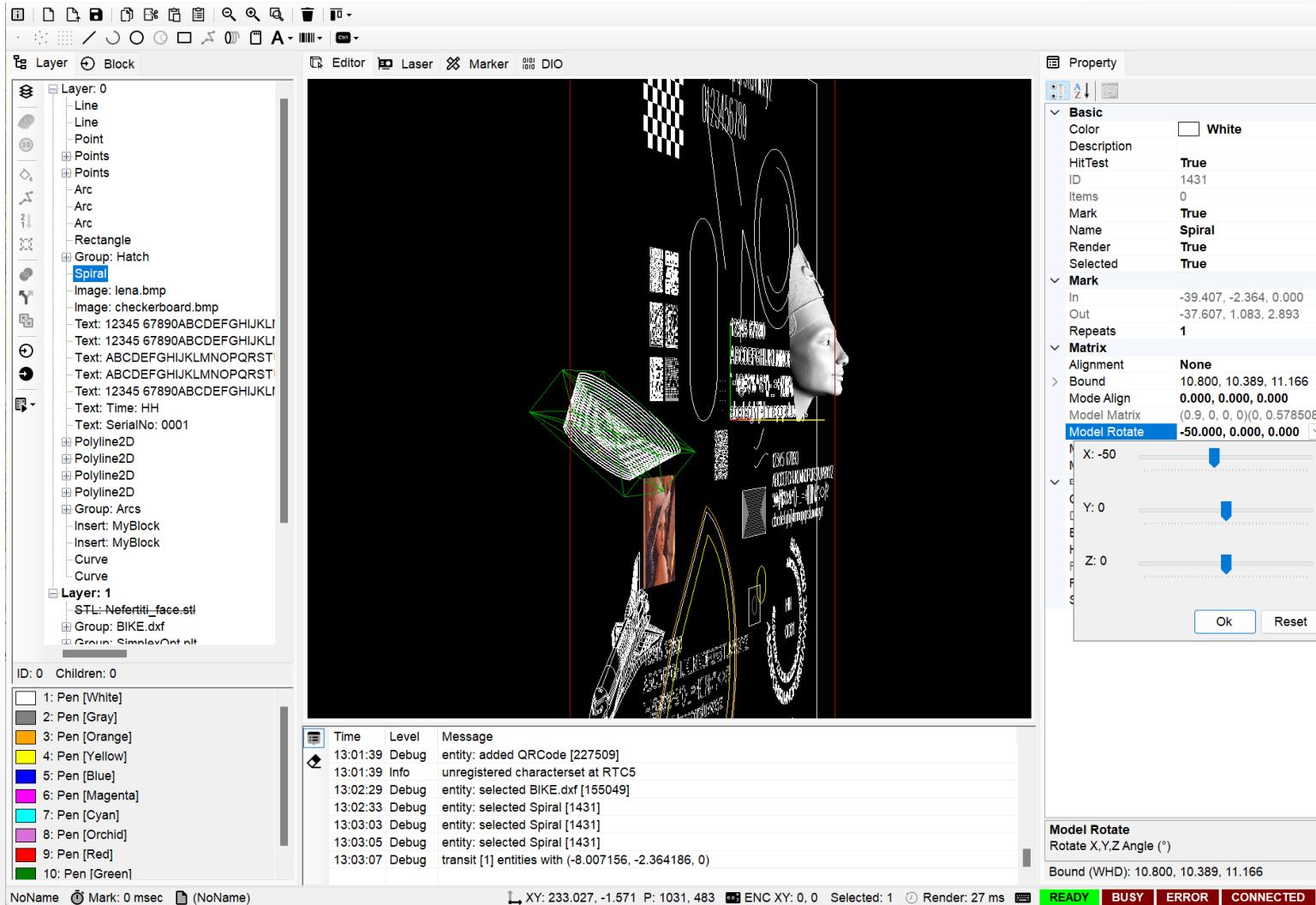
- Easy to integrate own laser application
- Support SCANLAB RTC4,5,6 and XL-SCAN(by syncAXIS) with multiple instances
- Support Powerful 3D render engine by OpenGL
- Support 2D, 3D scanner field correction
- Support unlimited vector data by automatically
- Support powerful options like as MoF, 2<sup>nd</sup> head, 3D, Ramp, sky writing, ...
- Support measurement scanner trajectory and plot to graph
- Support powerful C# script engine
- Support many kinds of laser sources to control
- Support many kinds of powermeters
- Multiple demo projects for easy to use

# 3D Editor



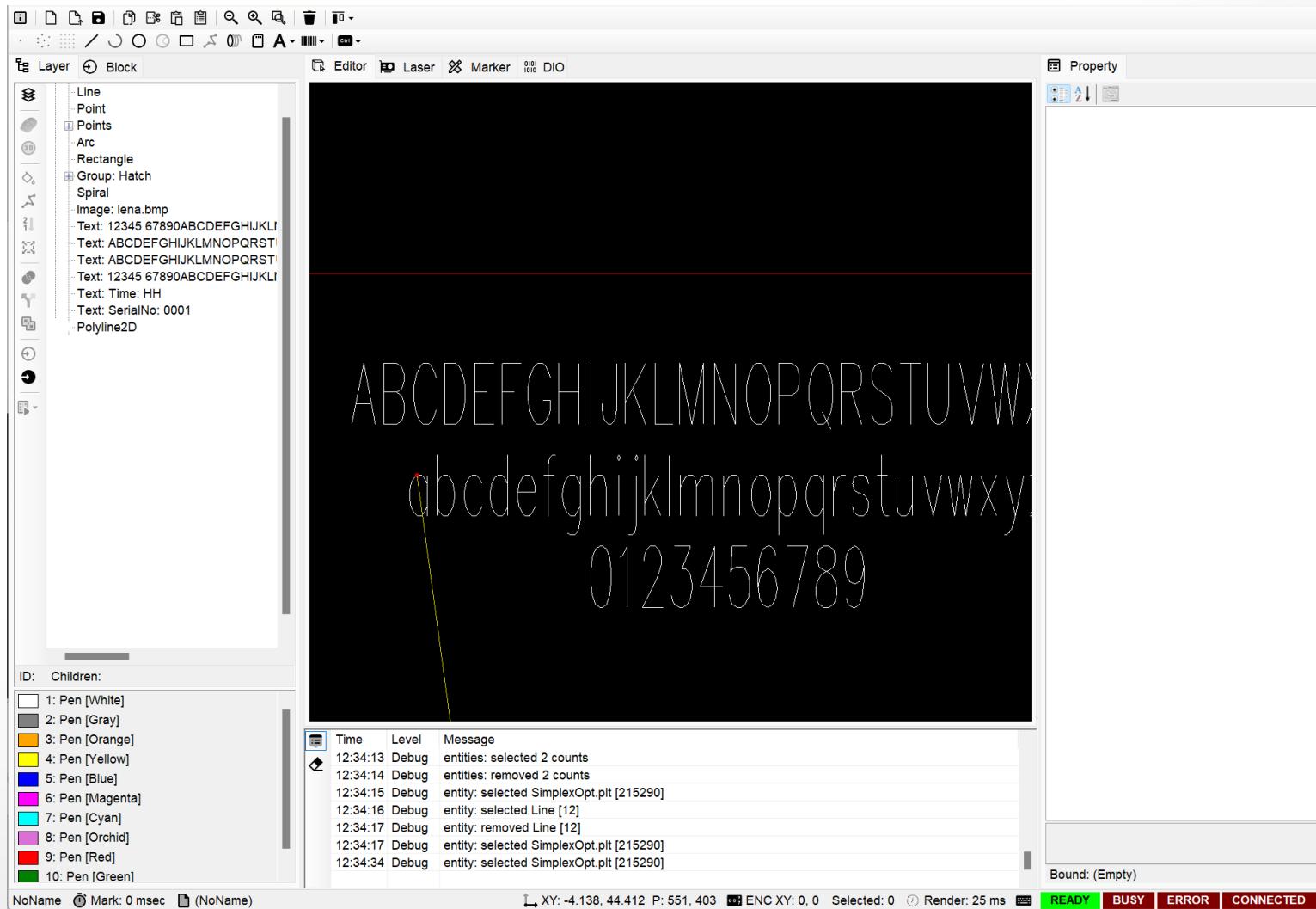
- Powerful 3D render engine by OpenGL
- Perspective view with free camera
- Show list of entities at Treeview
- Configurable pen parameters
- Editable entity properties
- Integrated RTC and laser control
- Customizable marker control
- Output log message

# Supported 4x4 Model Matrix



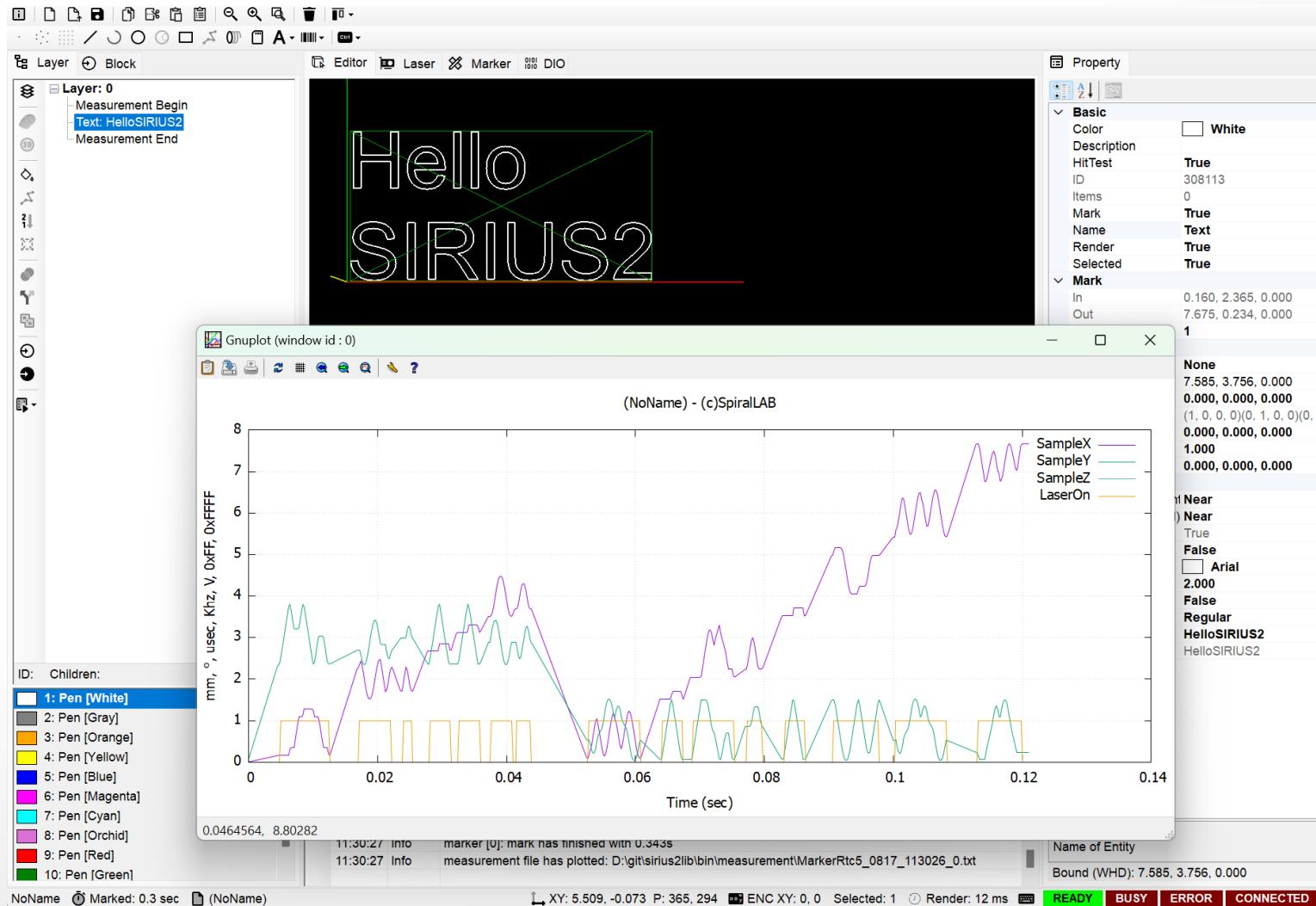
- Can be configure model rotate, transit and scale
- Free movable(or rotate by yaw, pitch) camera
- Z value for shift laser focus location

# Simulation Mark



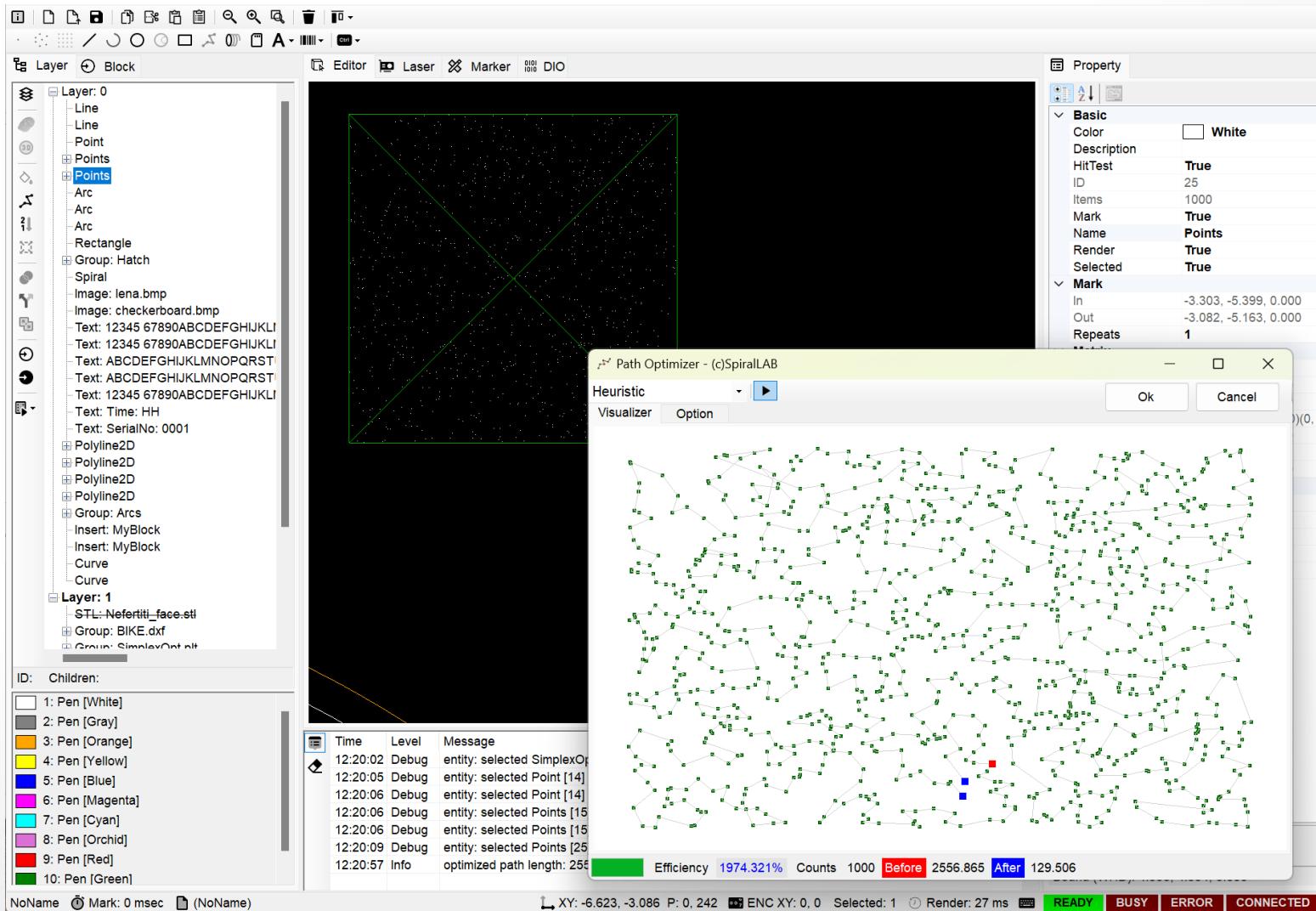
- Preview order of marks
- Selectable simulation speed by slow, normal and fast
- Also, followable look at camera

# Measurement



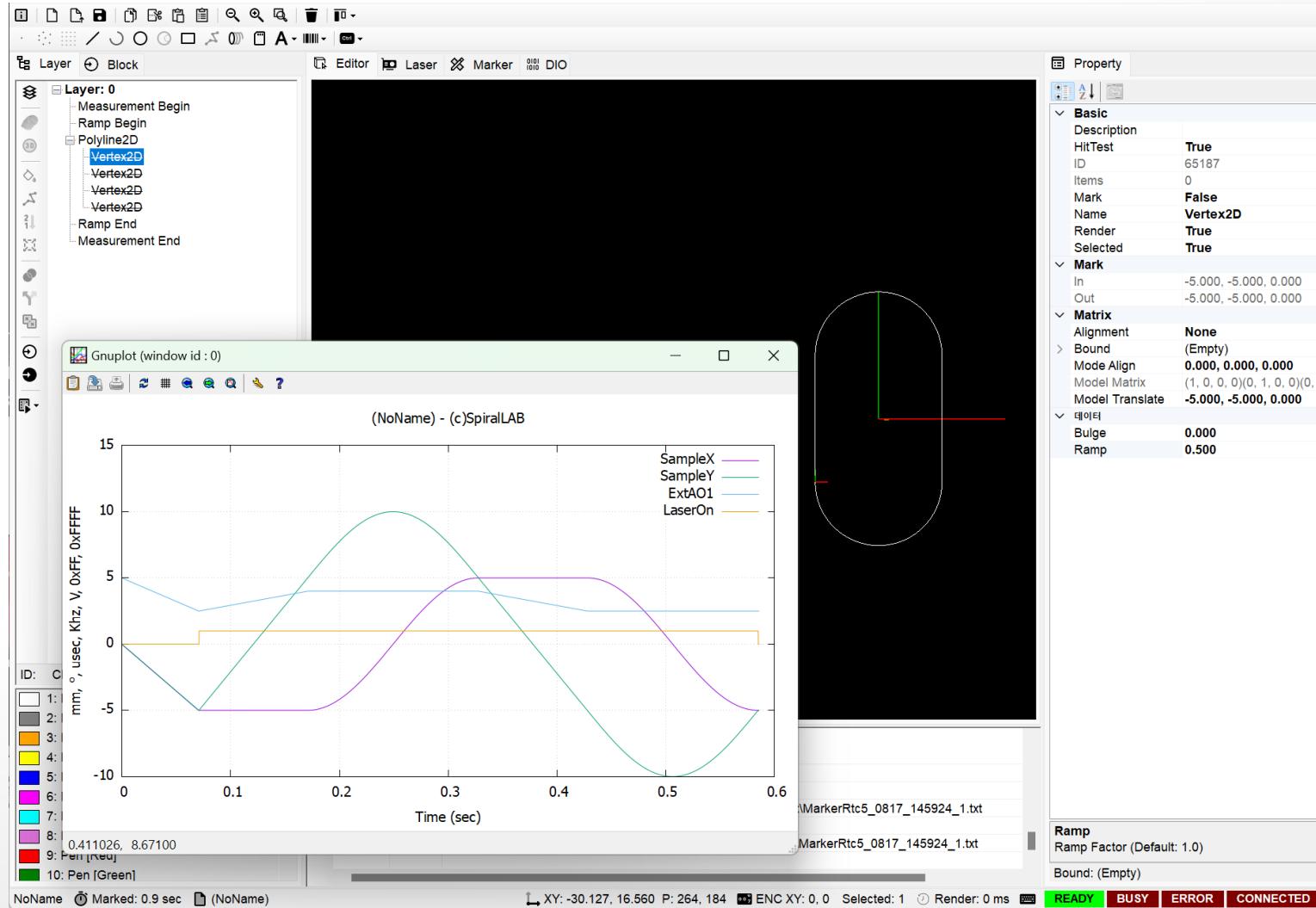
- Measurement scanner trajectory and laser signals
- Plot as graph by ‘gnuplot’
- Sampling up to 100KHz and gathering data with up to 8 channels
- For example, can do gather scanner z trajectory, external ENCODER values

# Path Optimizer



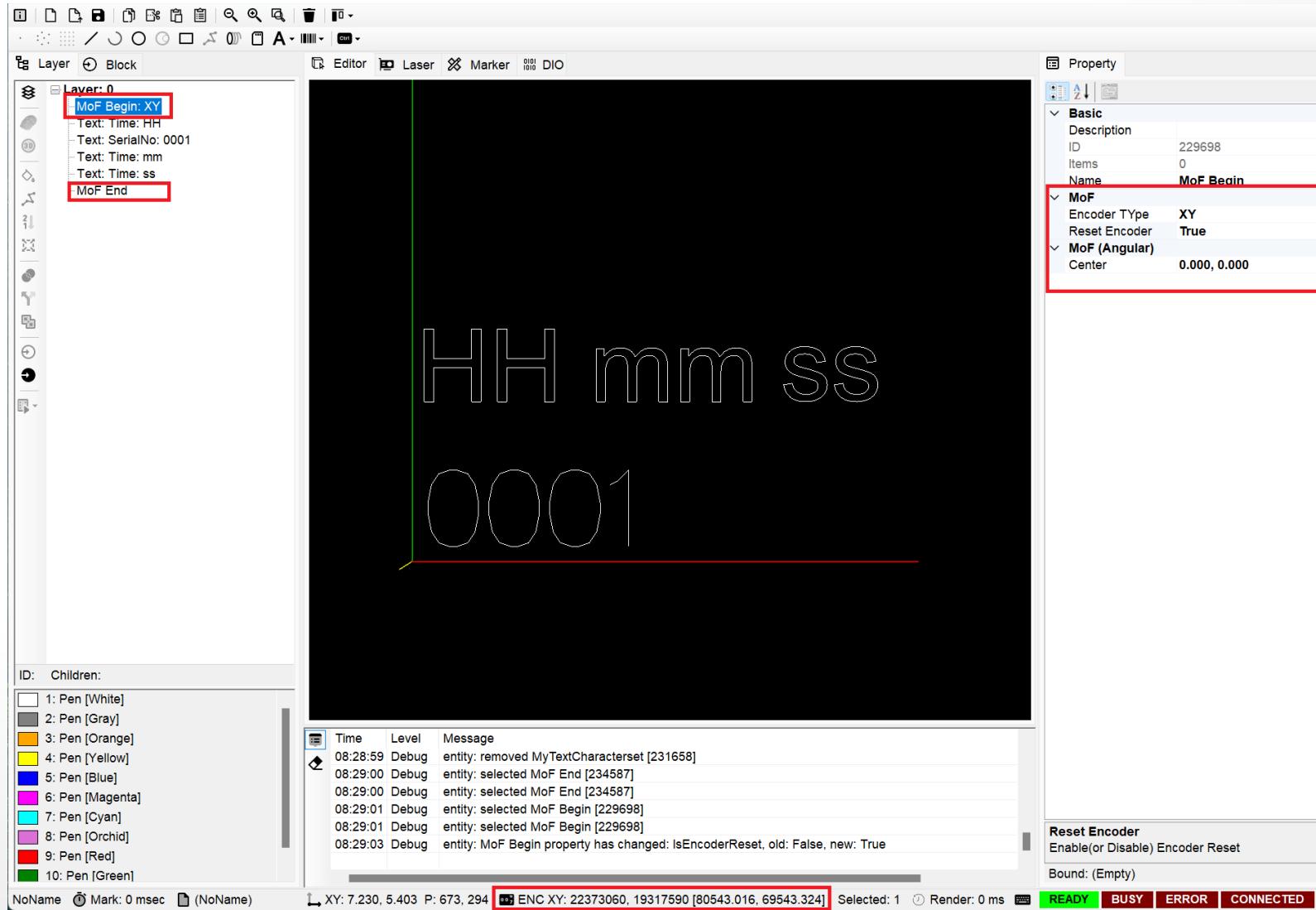
- Various ways to optimize path like as simple, heuristic, rotate, ...
- Path optimizer can be works combined with all child entities
- Supported reference positions : in, out or center

# Ramp(Automatic Laser Control)



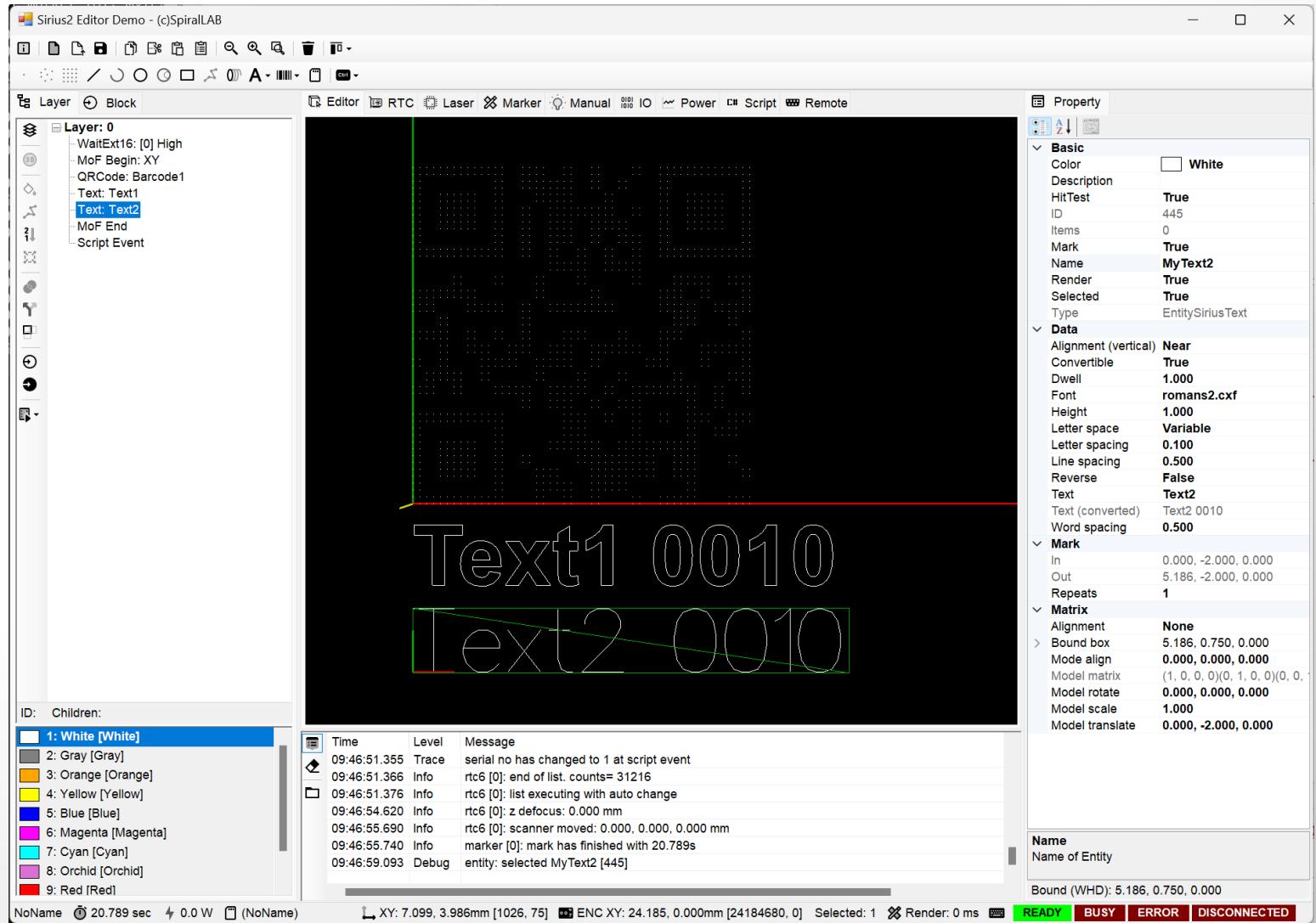
- ALC by defined vector
- Each vertex with ramp scale factor
- Example: Output Analog1
- Starting: 5V
- First vertex: 2.5V = (X0.5)

# XY(or angular) Marking on the fly



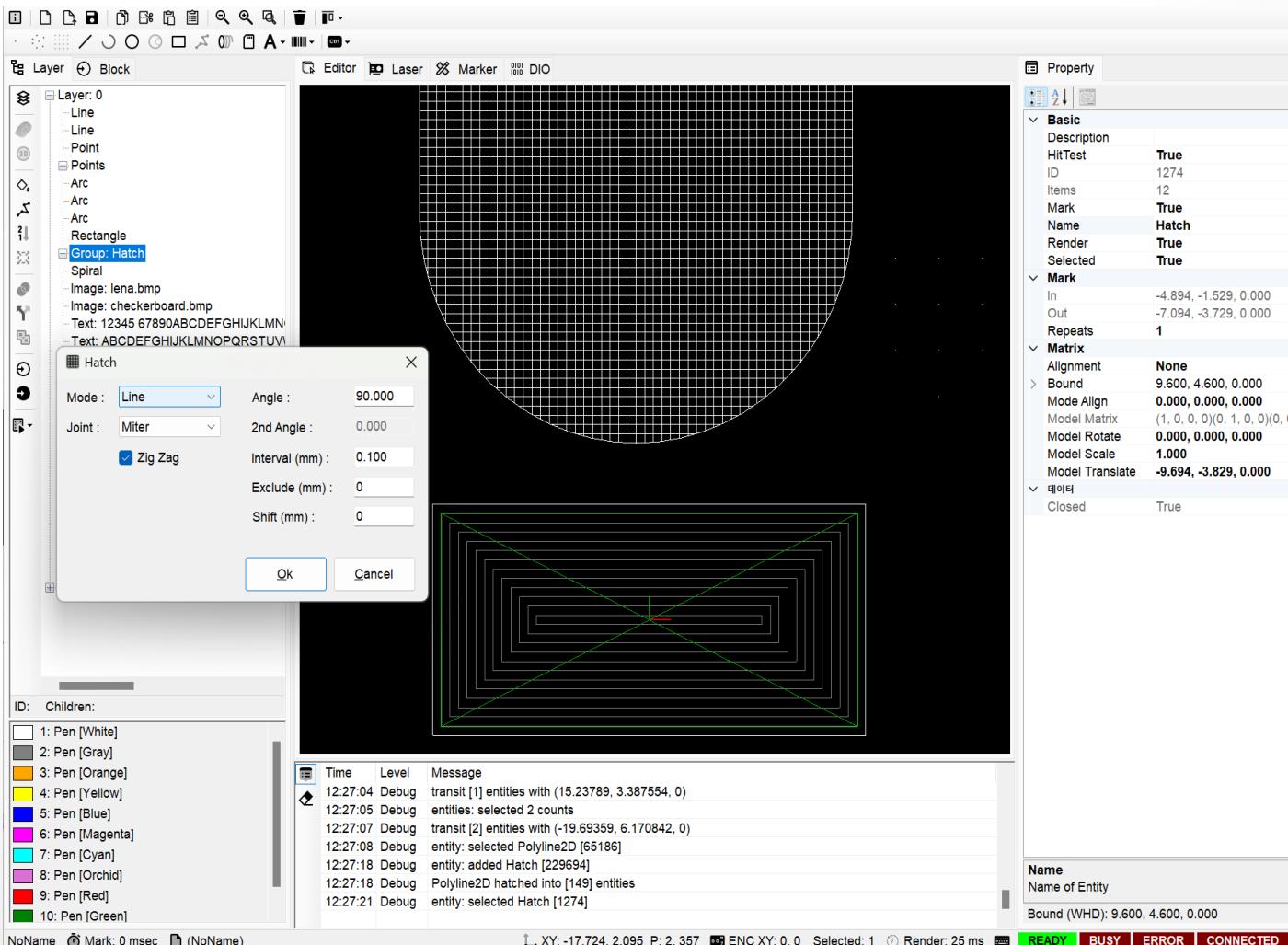
- MoF with XY Encoder
- MoF with Angular Encoder
- Wait for encoder by position (under, over or auto)

# Wait trigger and MoF start (Example)



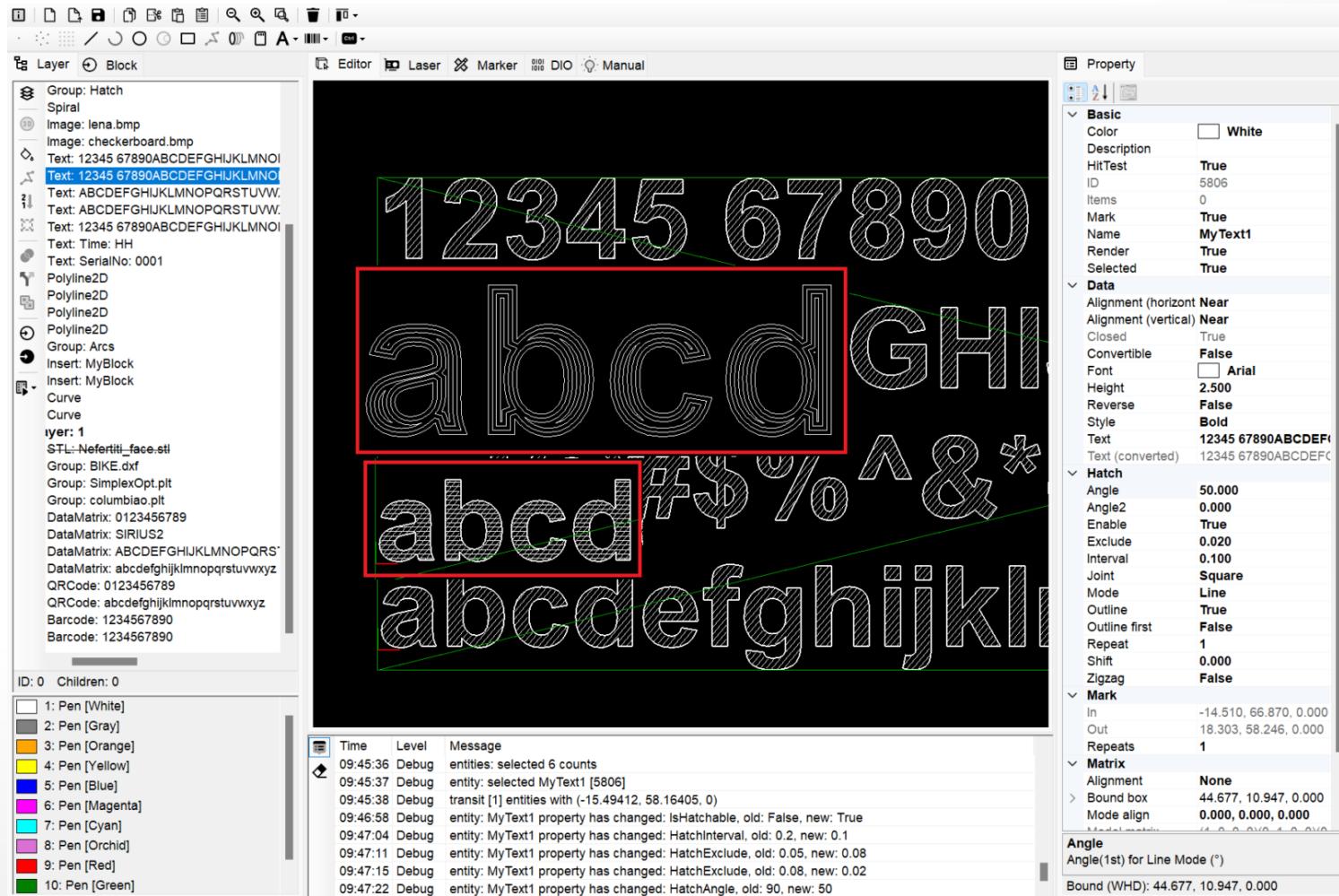
- Waiting trigger from D.IN0 at EXTENSION1 PORT goes to rising edge

# Extract Hatch Pattern



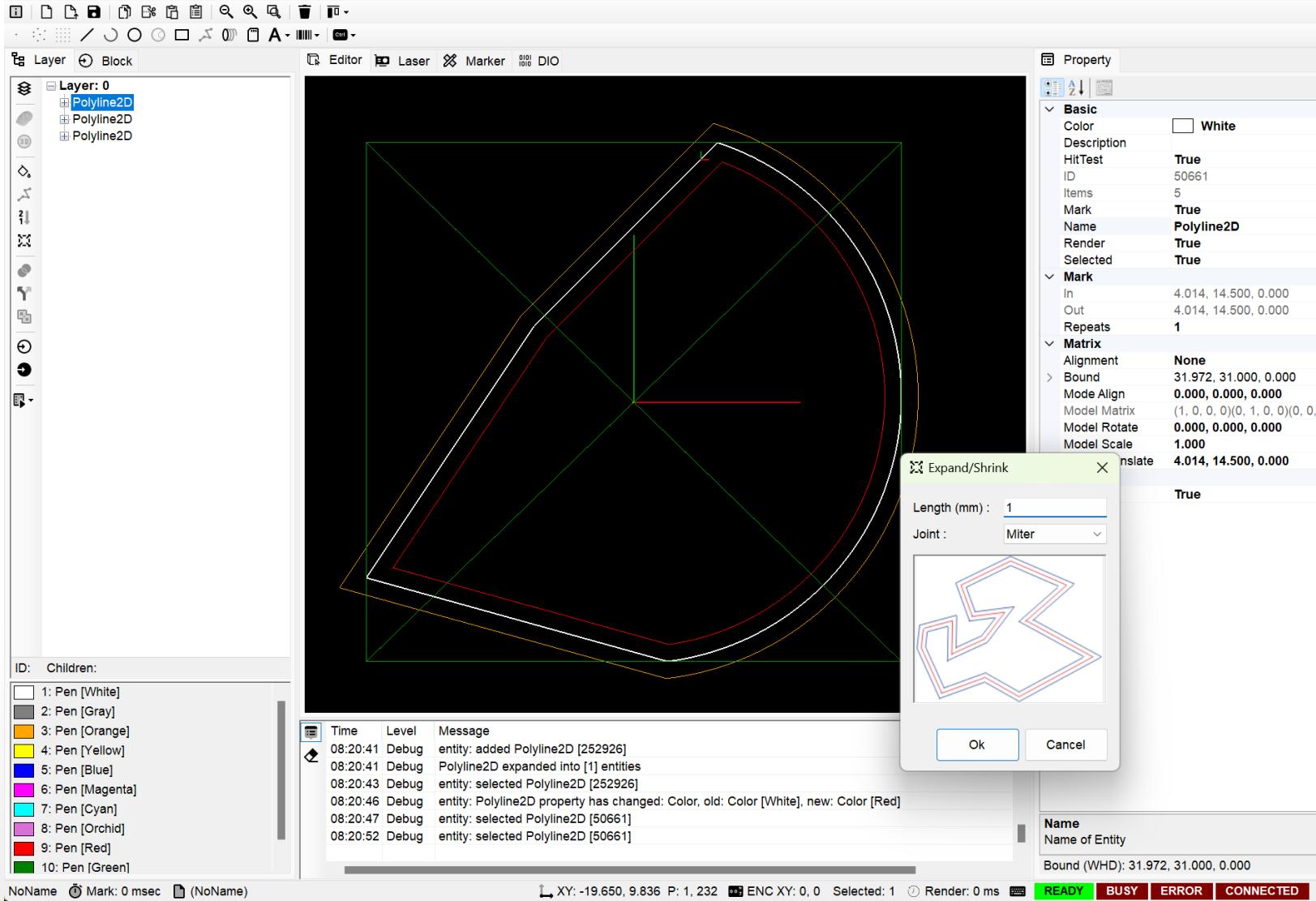
- Closed path are support to do hatch.
- Supported mode are line, cross line and polygon
- Editable hatch angle, interval, exclude length also.

# Text with Hatch



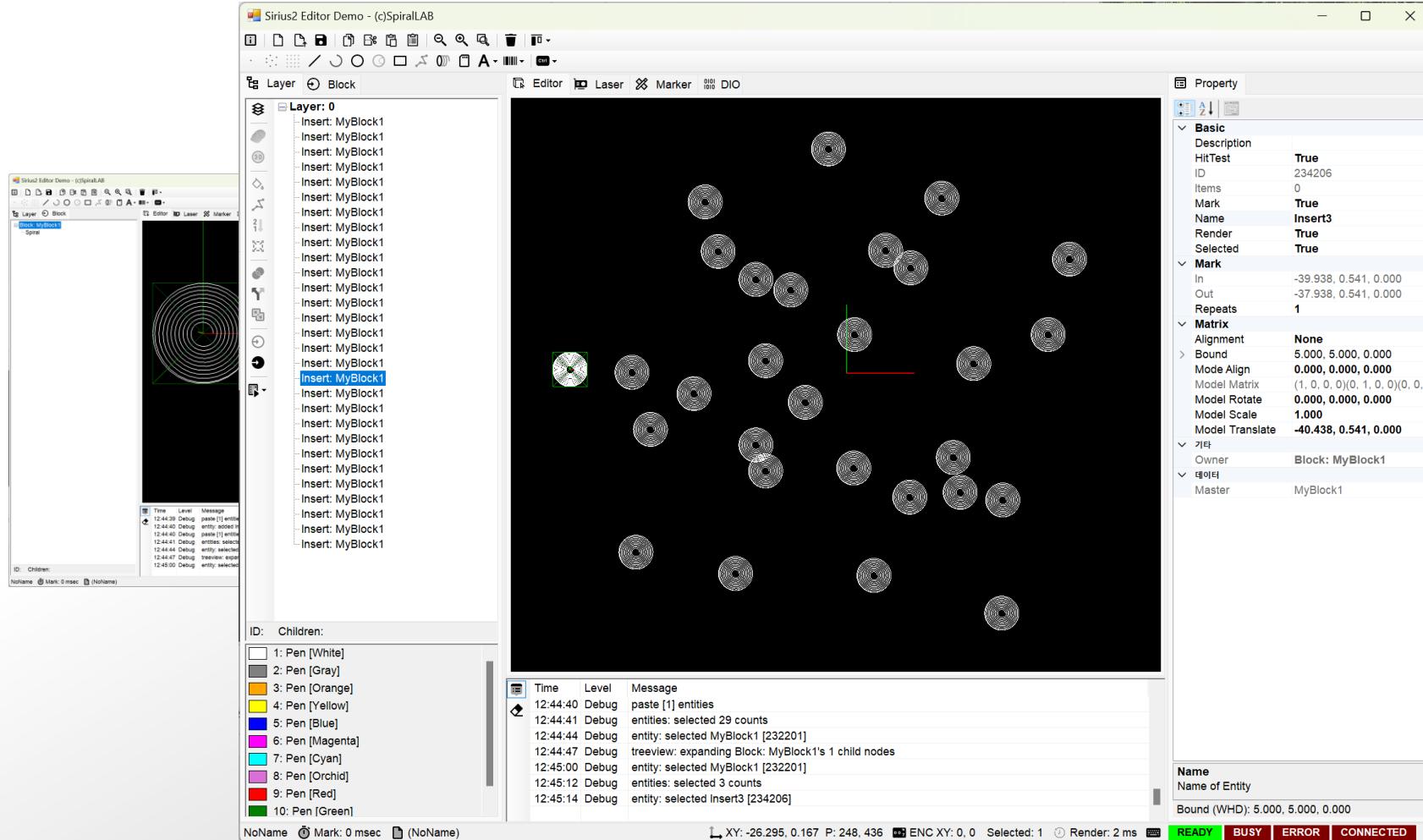
- Closed figures like as Text are support various hatch patterns

# Expand/Shrink



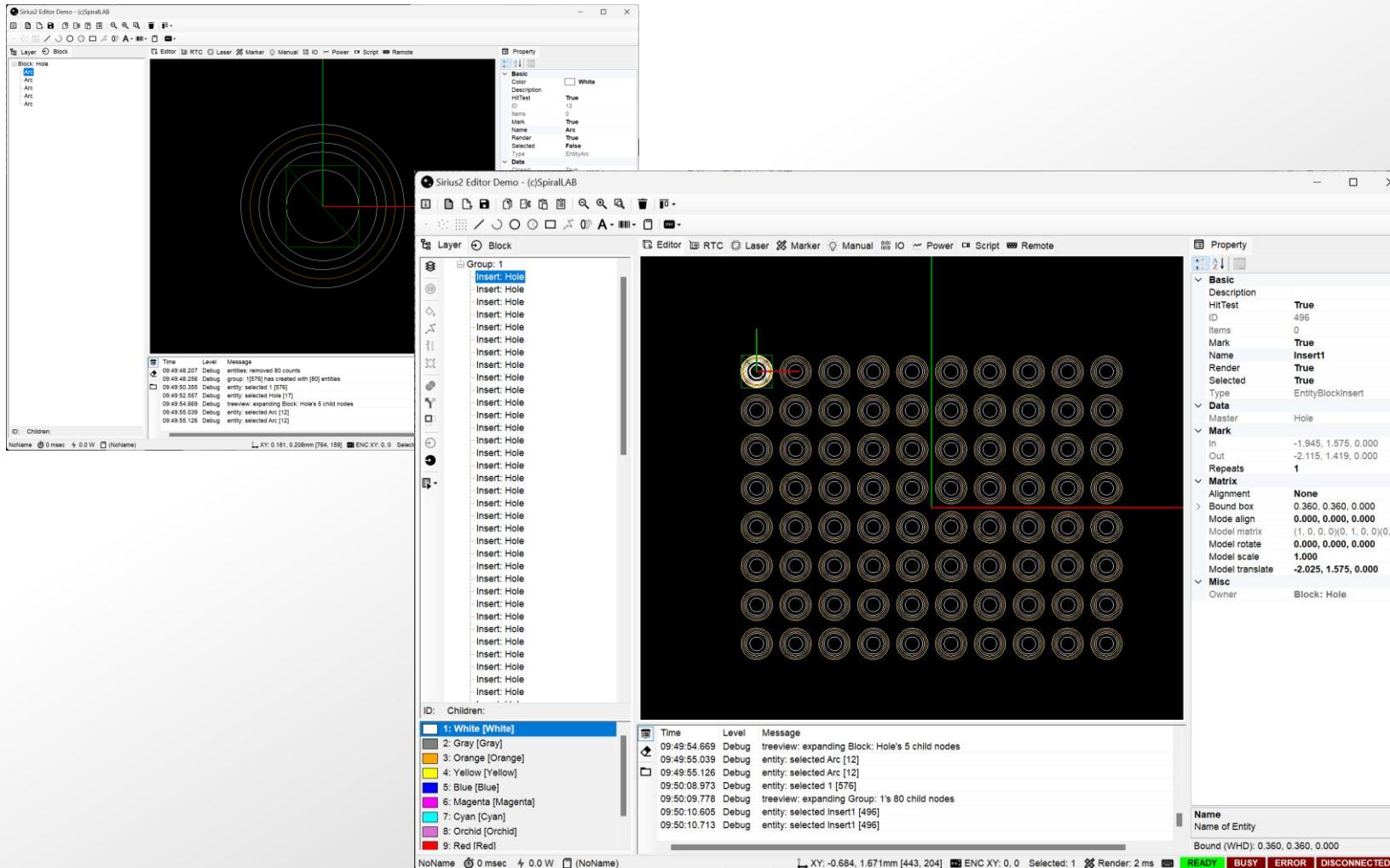
- Closed path are support to do expand/shrink
- To compensate laser spot size (also, bite or tool size compensate)

# Block and BlockInsert



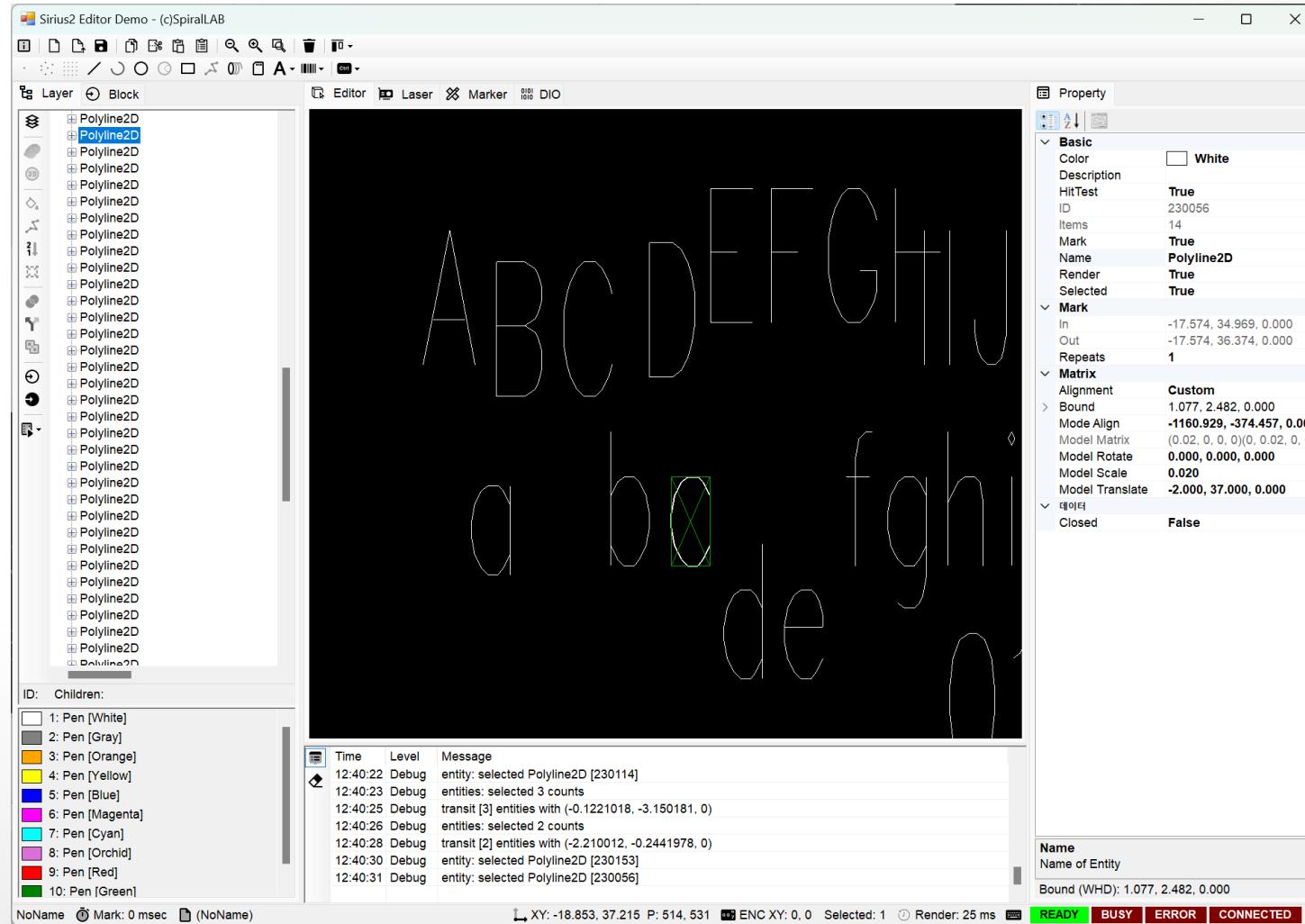
- Block and BlockInsert are supported
- Can be created as master block by user
- Can be inserted block also

# Block and BlockInsert (Example)



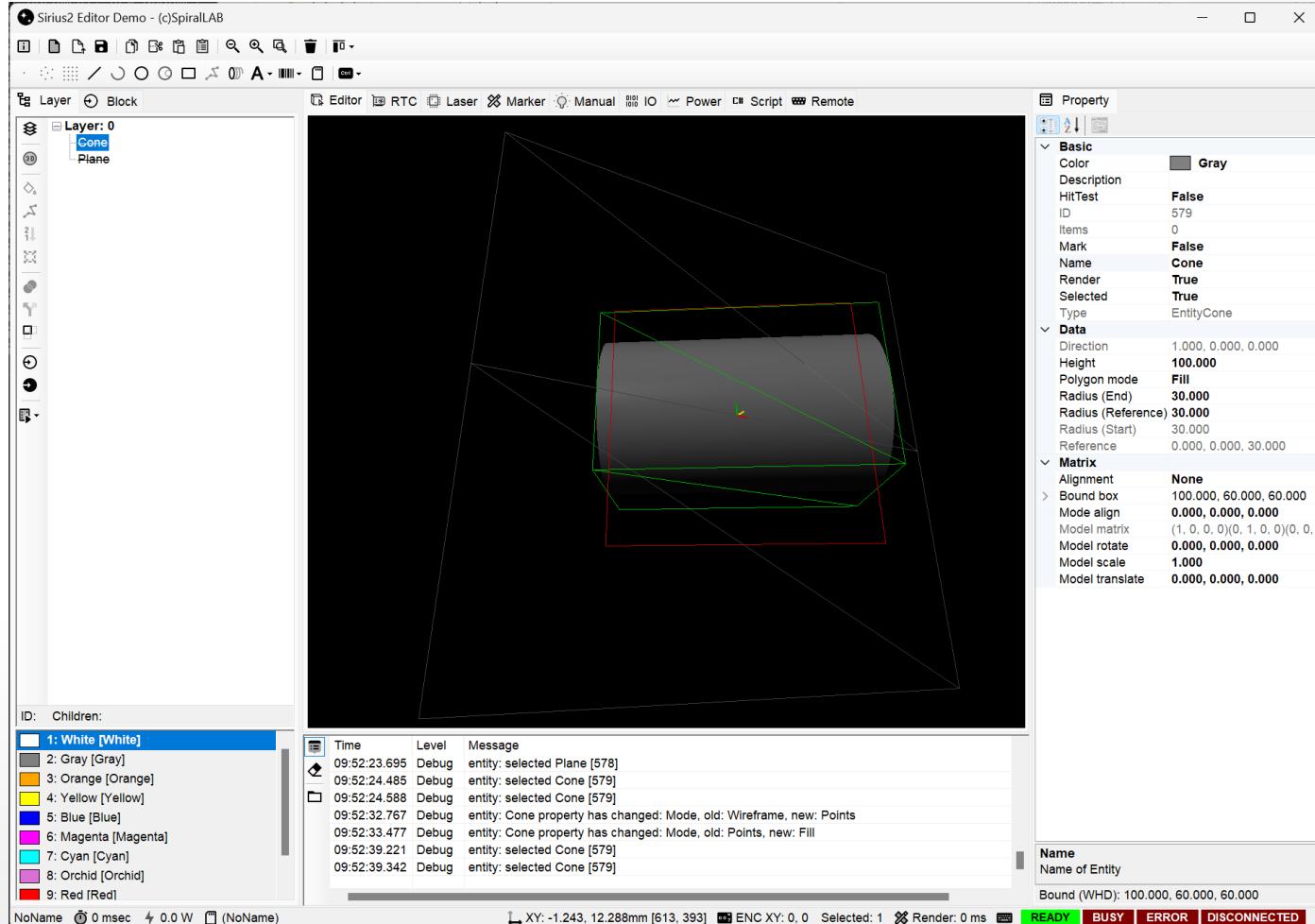
- Block insert with offset(x,y,z and anglez)
- PoP(Package on package) by laser drilling (aka. TSV)
- Editable each ‘Hole’
- Create array of holes

# Explode to Primitive Type



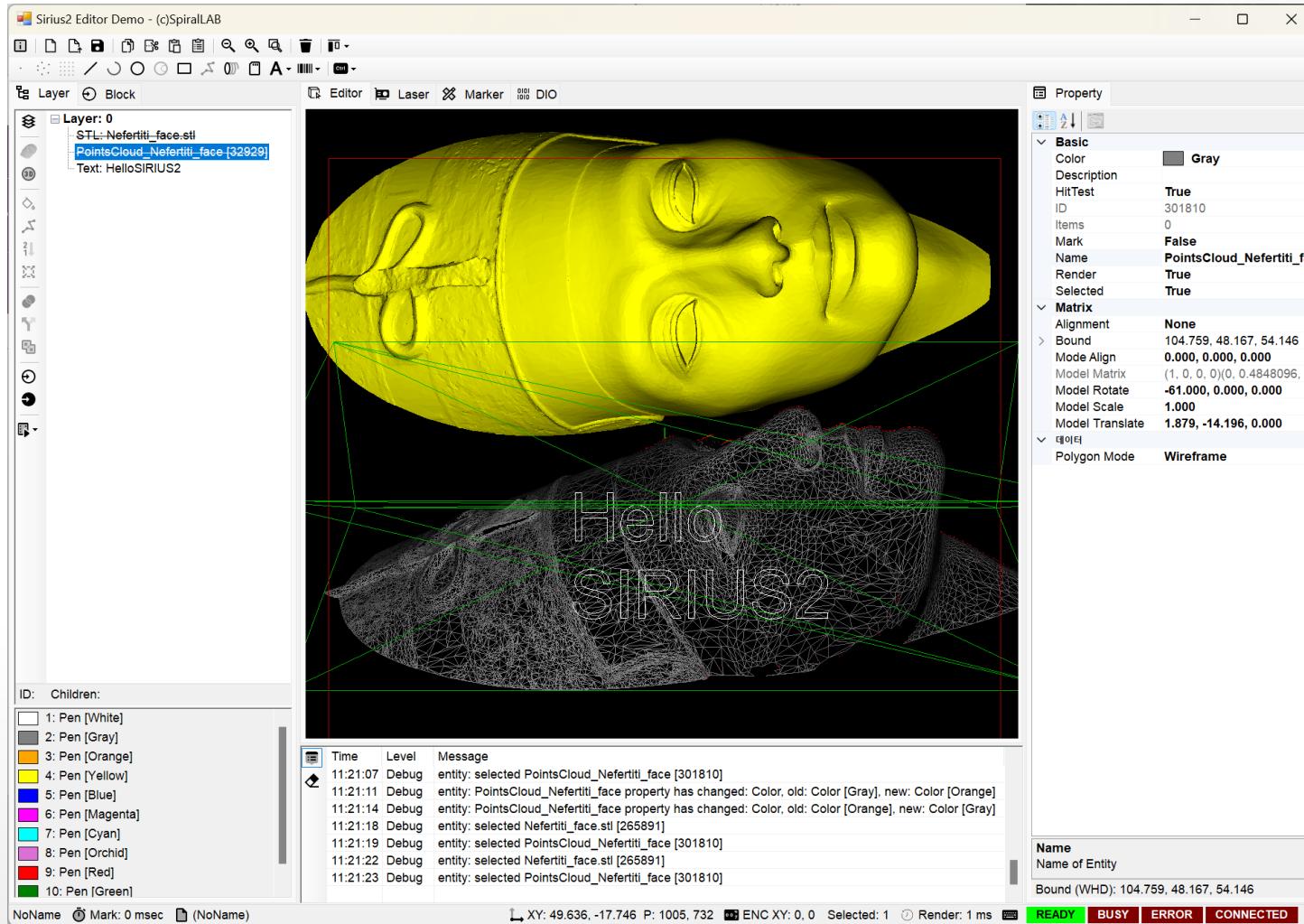
- Explode entity into primitive type
- Example: Group-> Polylines  
→ Line and arc

# 3D Field Correction



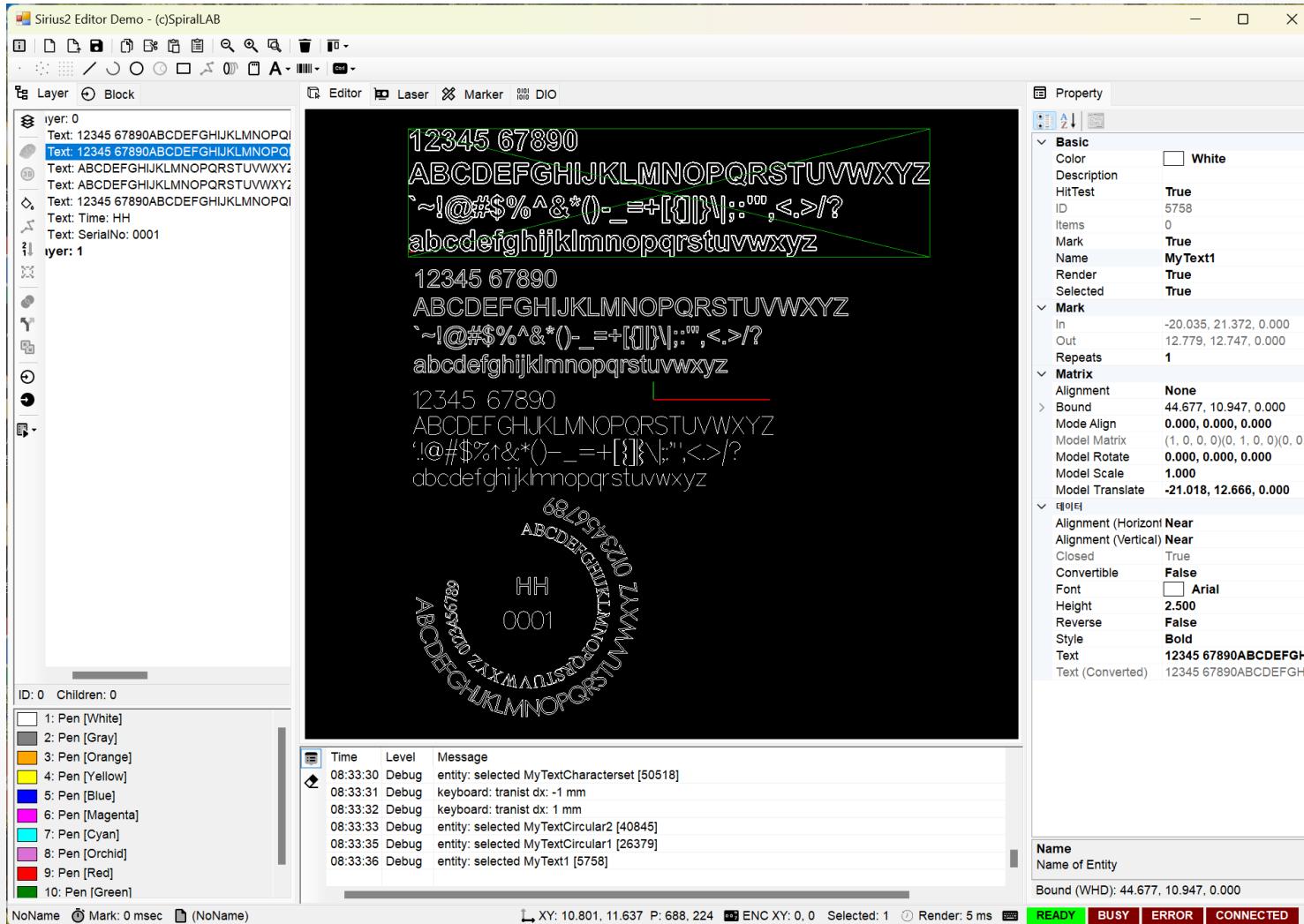
- Plane, Cylinder and Cone shapes
- Apply new 3d correction file and mark 2d vector data at 3d surface easily

# 3D Field Correction



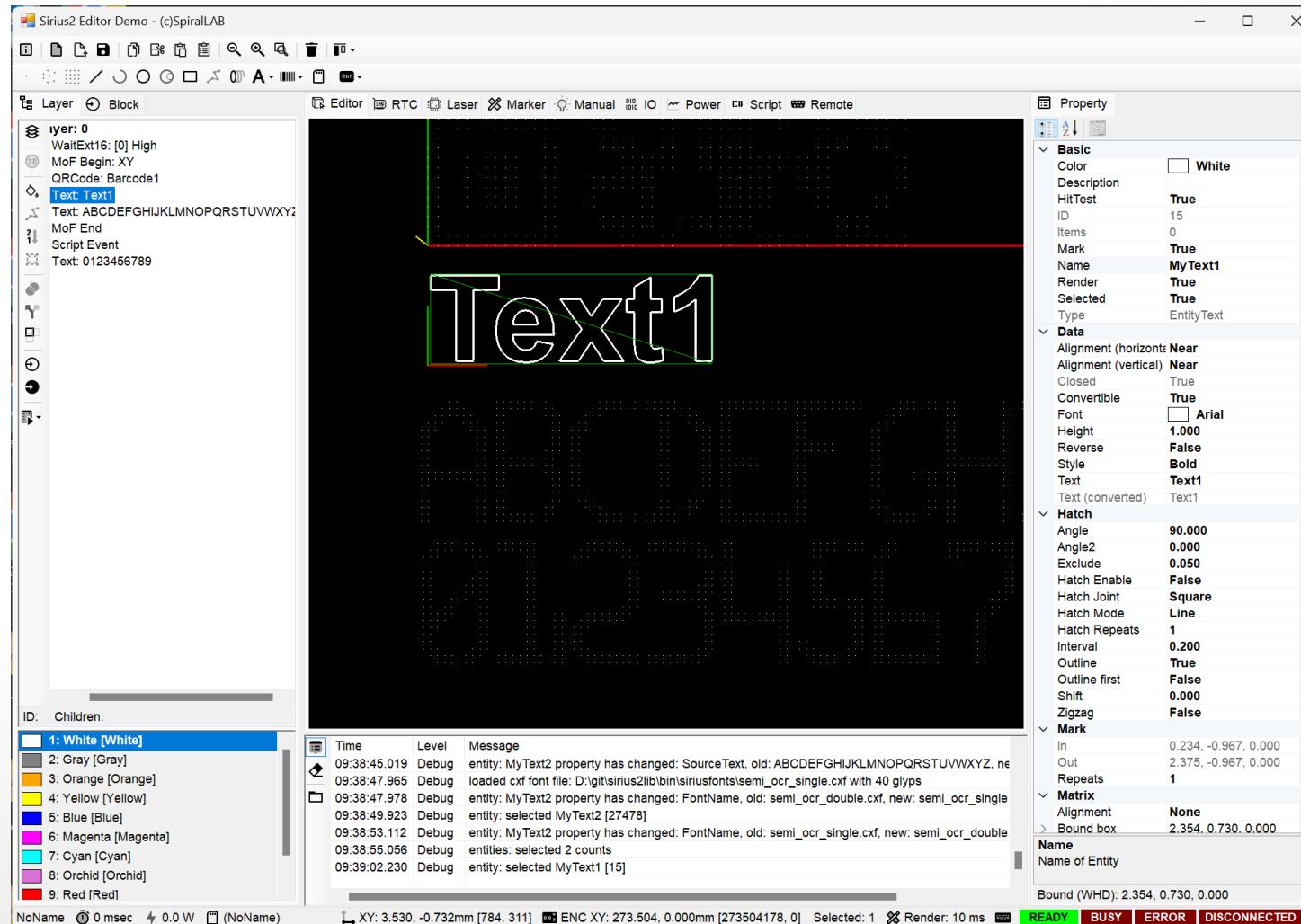
- Extract points cloud from surface 3D model and convert new 3d correction file
- Apply new 3d correction file and mark 2d vector data at 3d surface easily

# Various Fonts



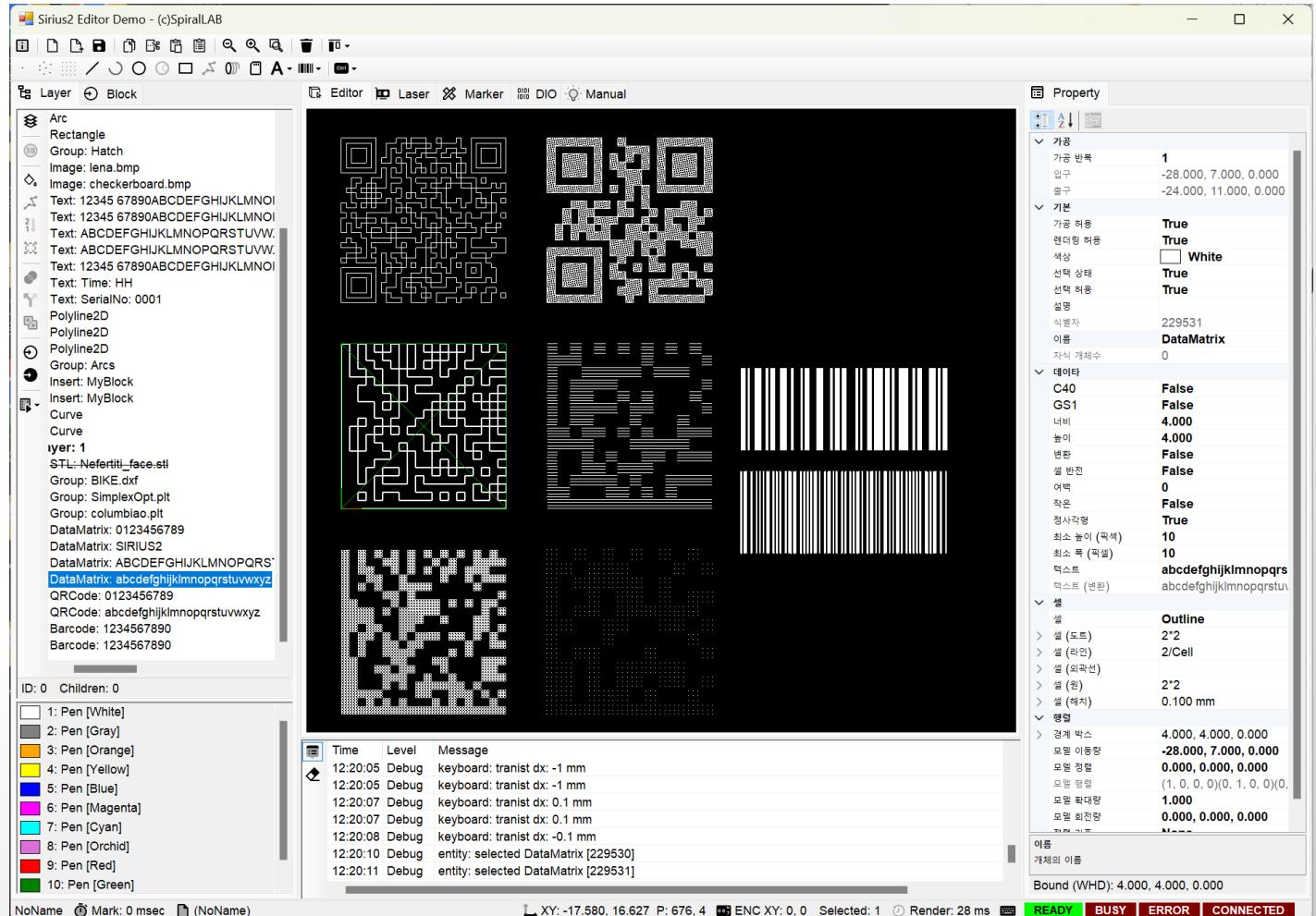
- **Text:** installed font files like as ttf
- **ImageText:** mark by rasterized bitmap
- **SiriusText:** open sourced font file like as cxf, lff
- **Circular text**
- **Characterset text :** register(or download) into RTC (used by date, time, serial no)

# SEMI OCR Font



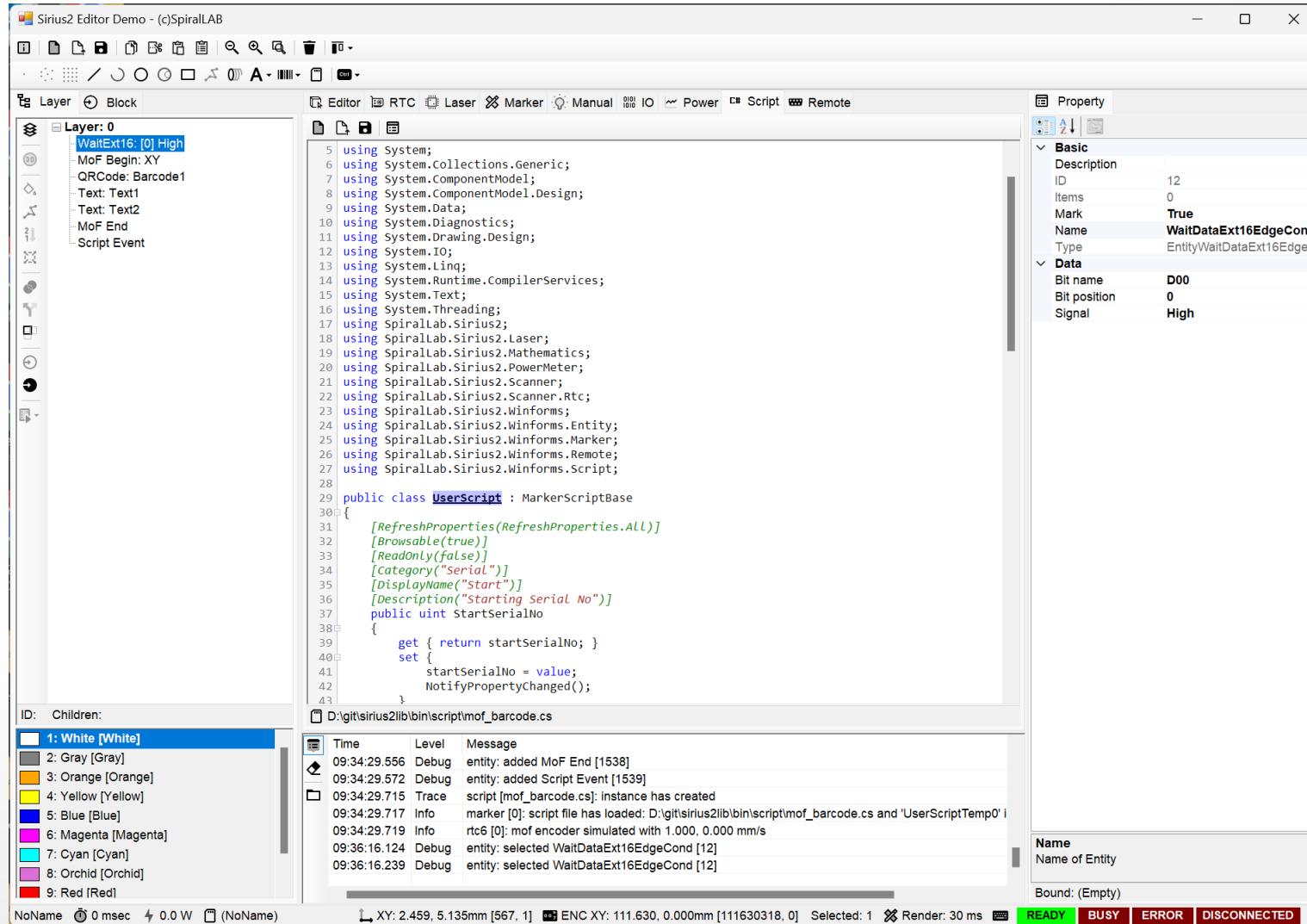
- Specific font for SEMI OCR Font
- Jump and shoot mode
- Mark each pixels during specific time(or dwell time)

# Various Barcode Cells



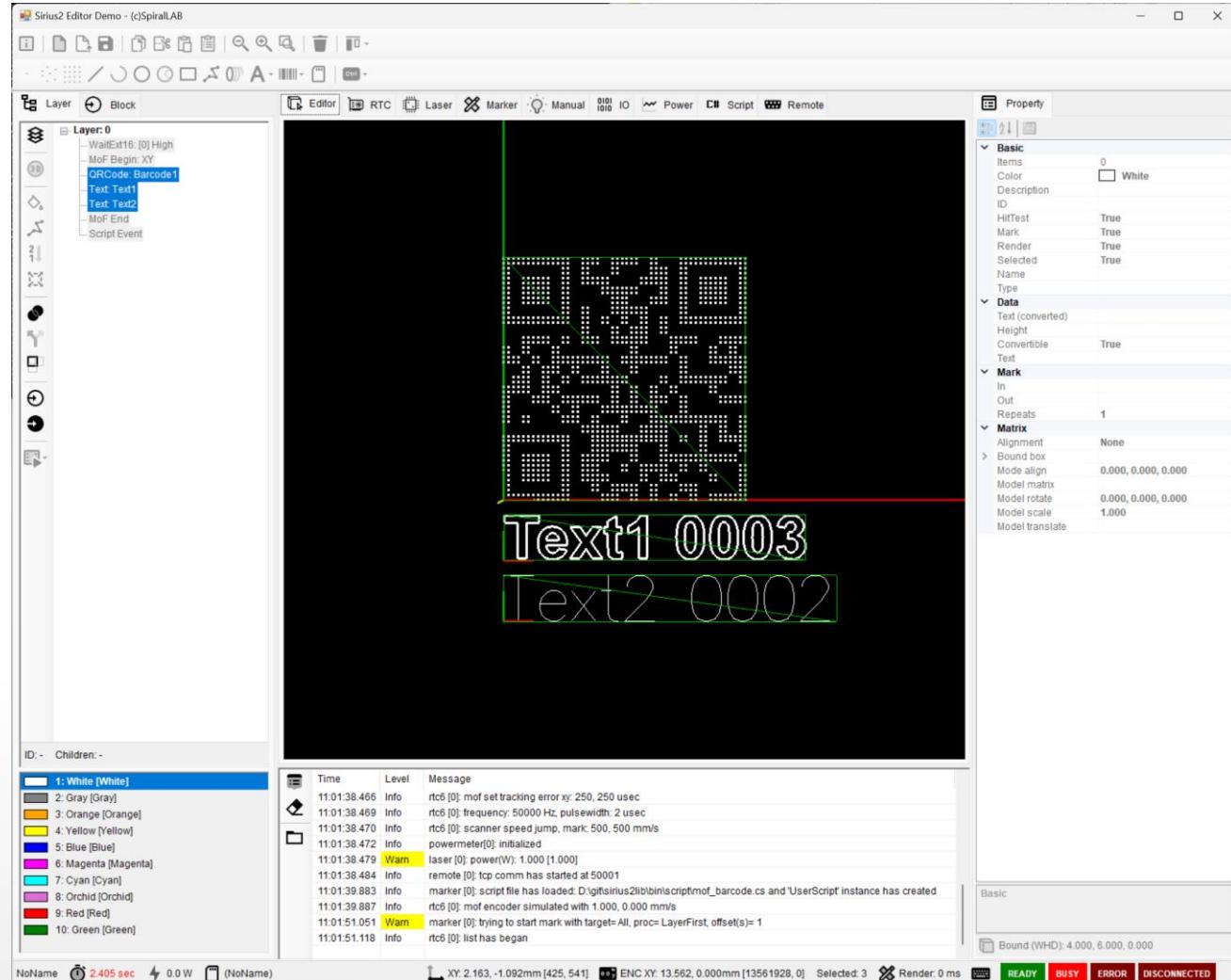
- Various barcode cell types : dots, lines, outline, circles and hatch
- DataMatrix, QRcode, PDF417 and 1D barcodes
- Barcode (also Text) data can be modified before mark at runtime with ease by event handler(or script function)

# C# Script



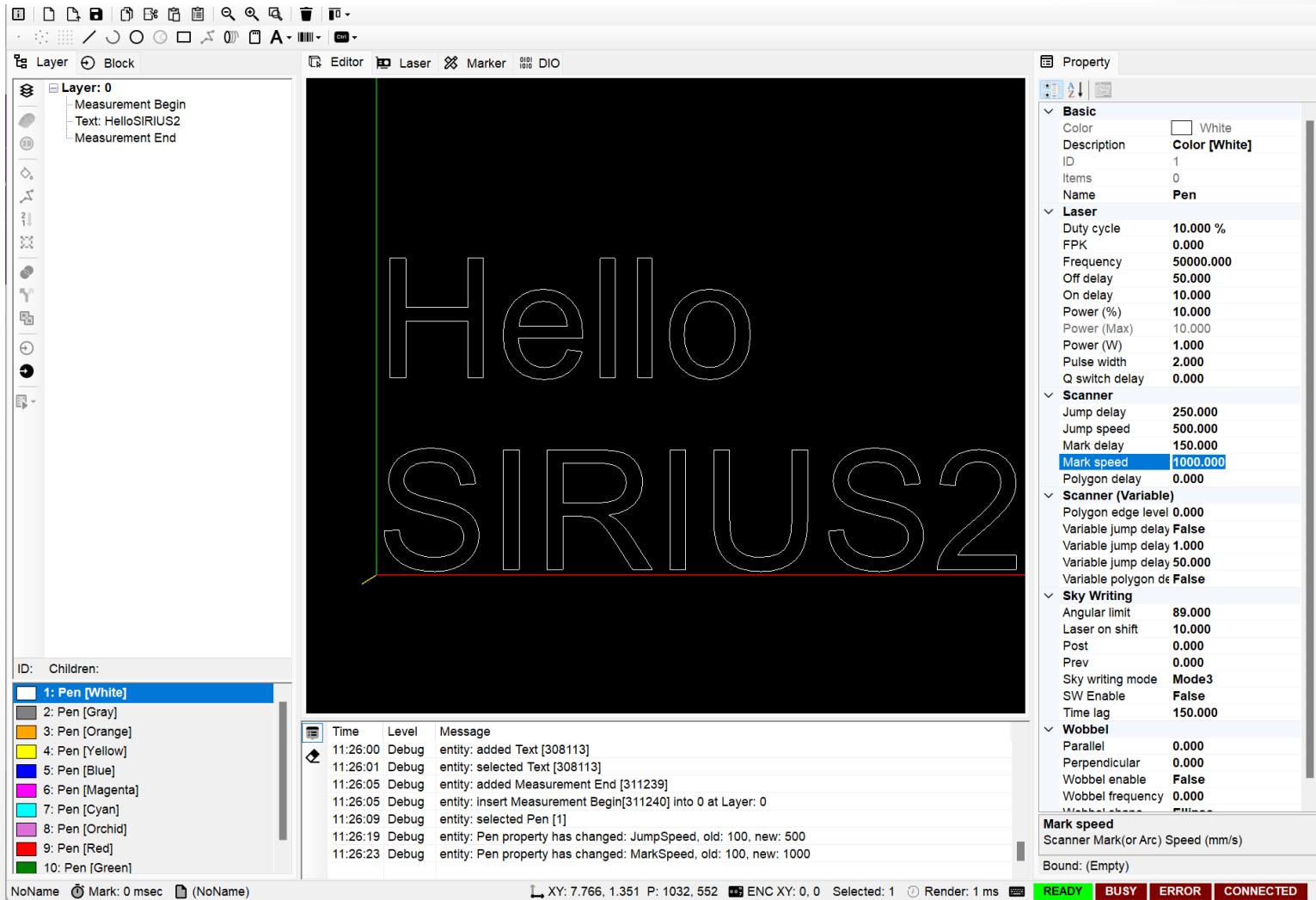
- Editable barcode and text entity data by C# script
- For customize user text formats at runtime
- User can do modify C# script codes at runtime
- For example: increase or reset serial number at any time

# C# Script with MoF + Barcode (Demo)



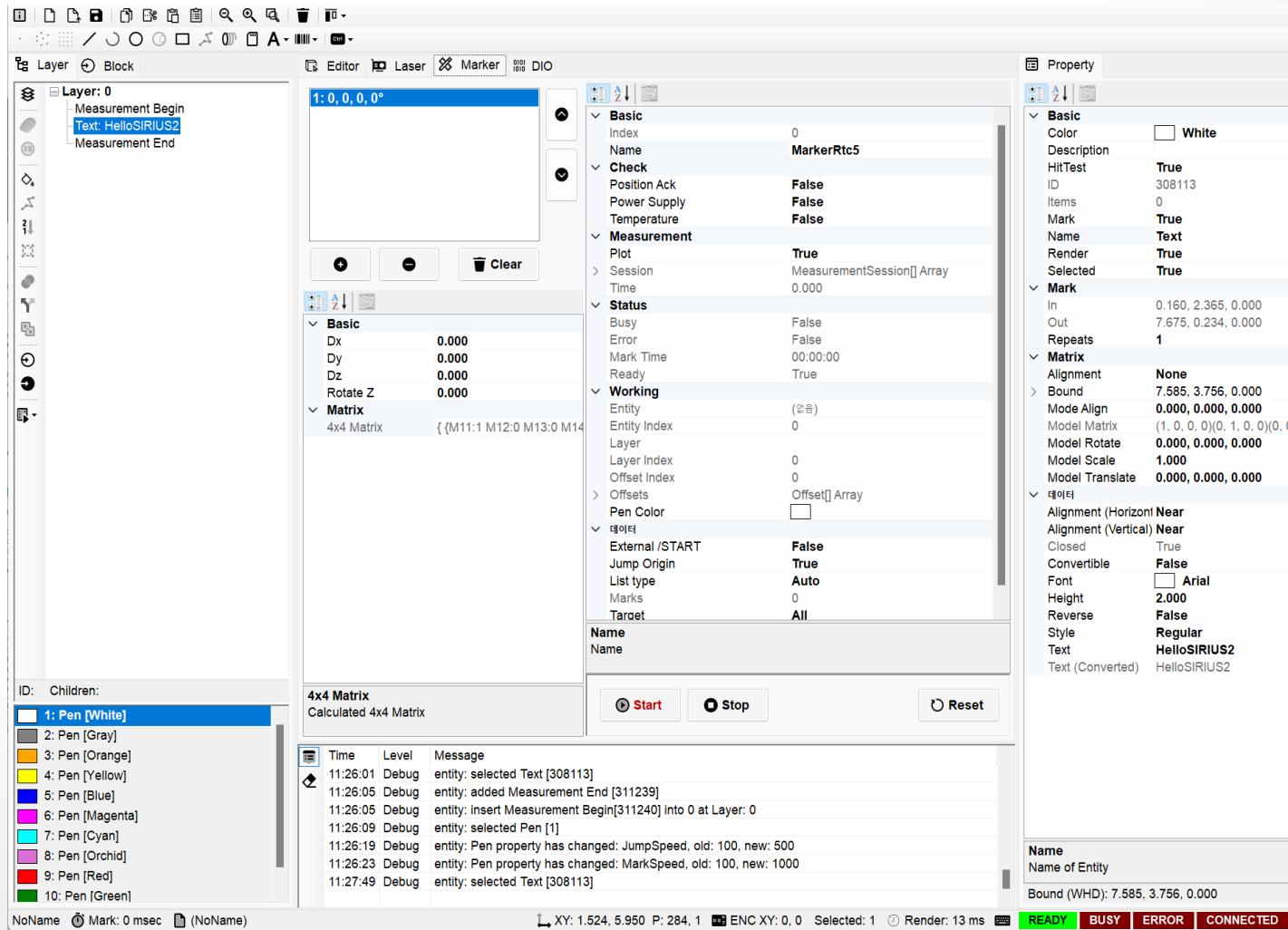
- Waiting external trigger : DIN0 at Extension Port
- Start x/y MoF with encoder reset
- Modified barcode and text data by C# script at runtime
- For example: Increase serial no
- End MoF
- Repeats (at Layer entity)

# Pen Parameters



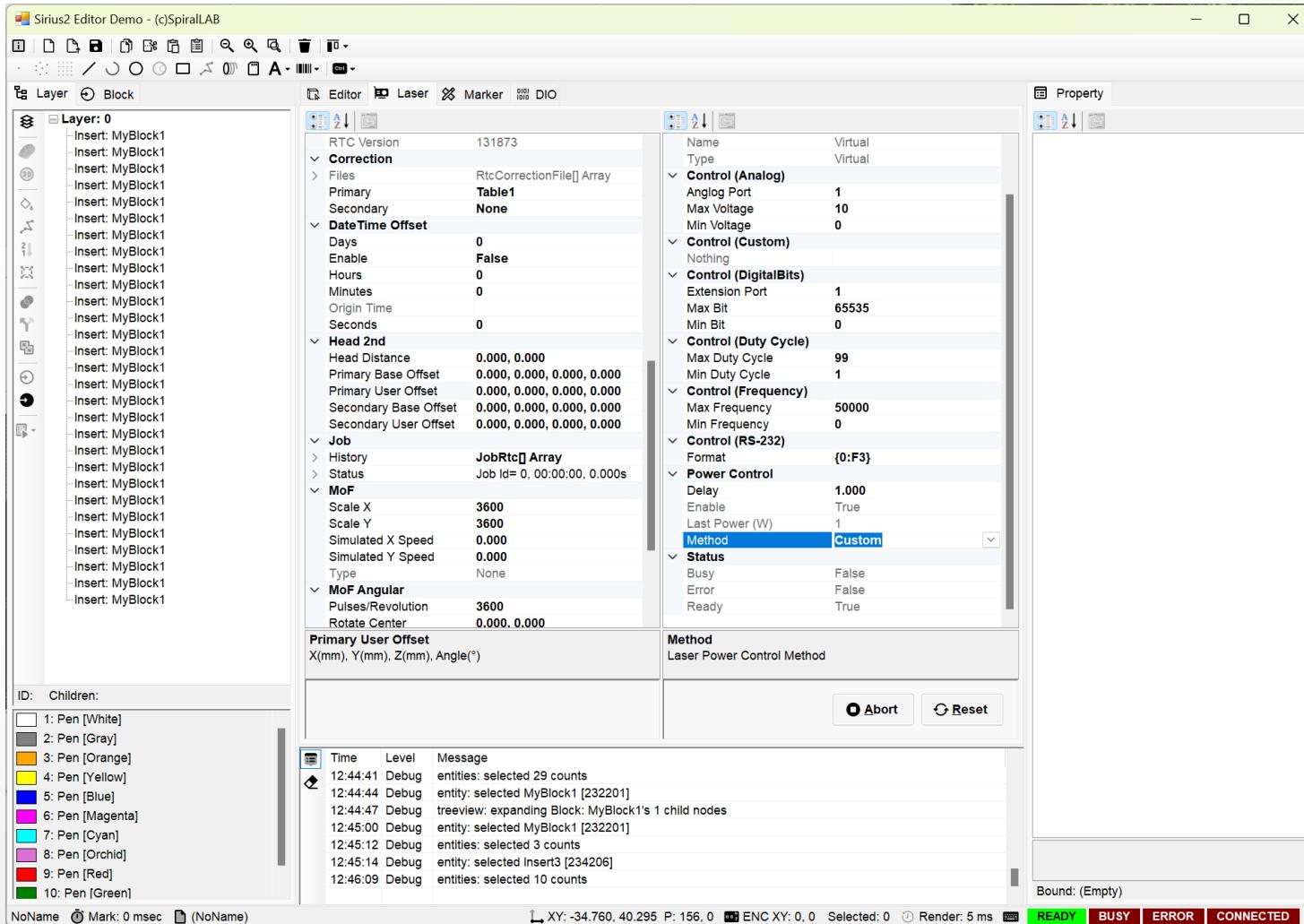
- Configurable pen parameters
- Editable frequency and pulse width
- Editable scanner speed and delays
- Editable output laser power
- Skywriting, wobble and mores

# Integrate Marker Control



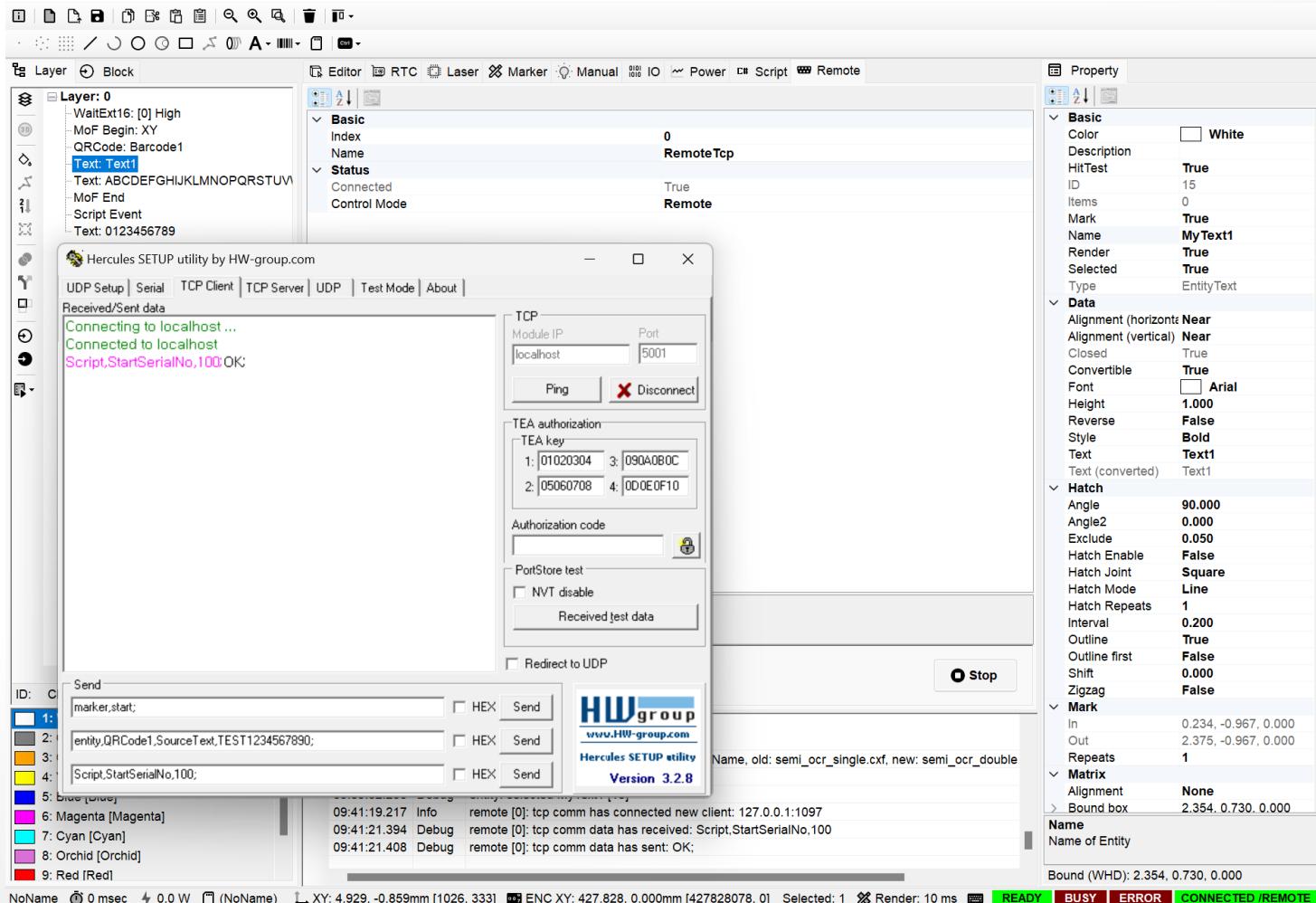
- Support multiple offset(dx,dy,dz and anglez) locations
- Ready for RTC marker
- Ready for syncAXIS marker
- Supported user customizable marker (open sourced)

# Integrated RTC and Laser Control



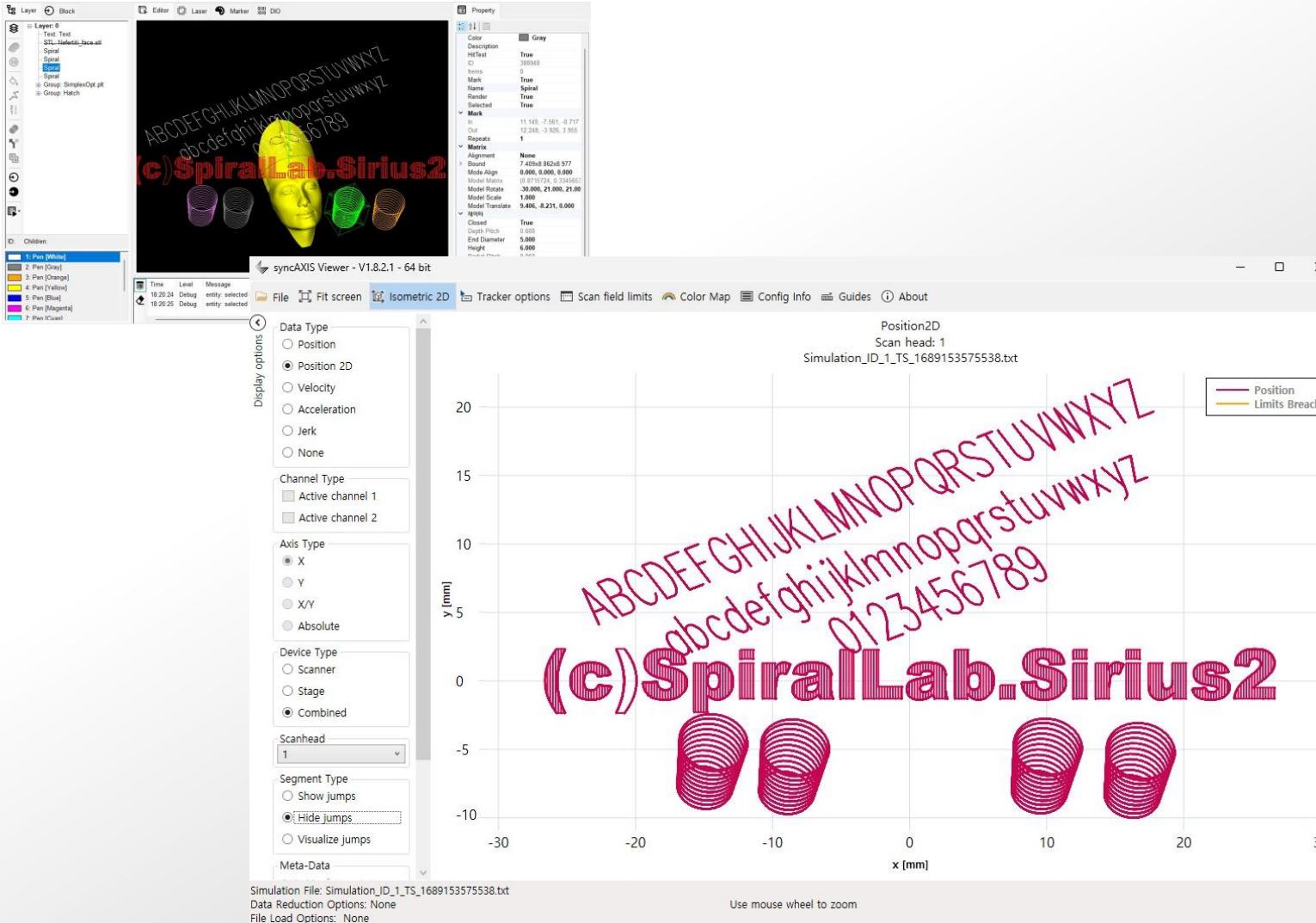
- Control RTC parameters
- Control output laser power by analog, pulse width, digital output, ...

# Remote Control



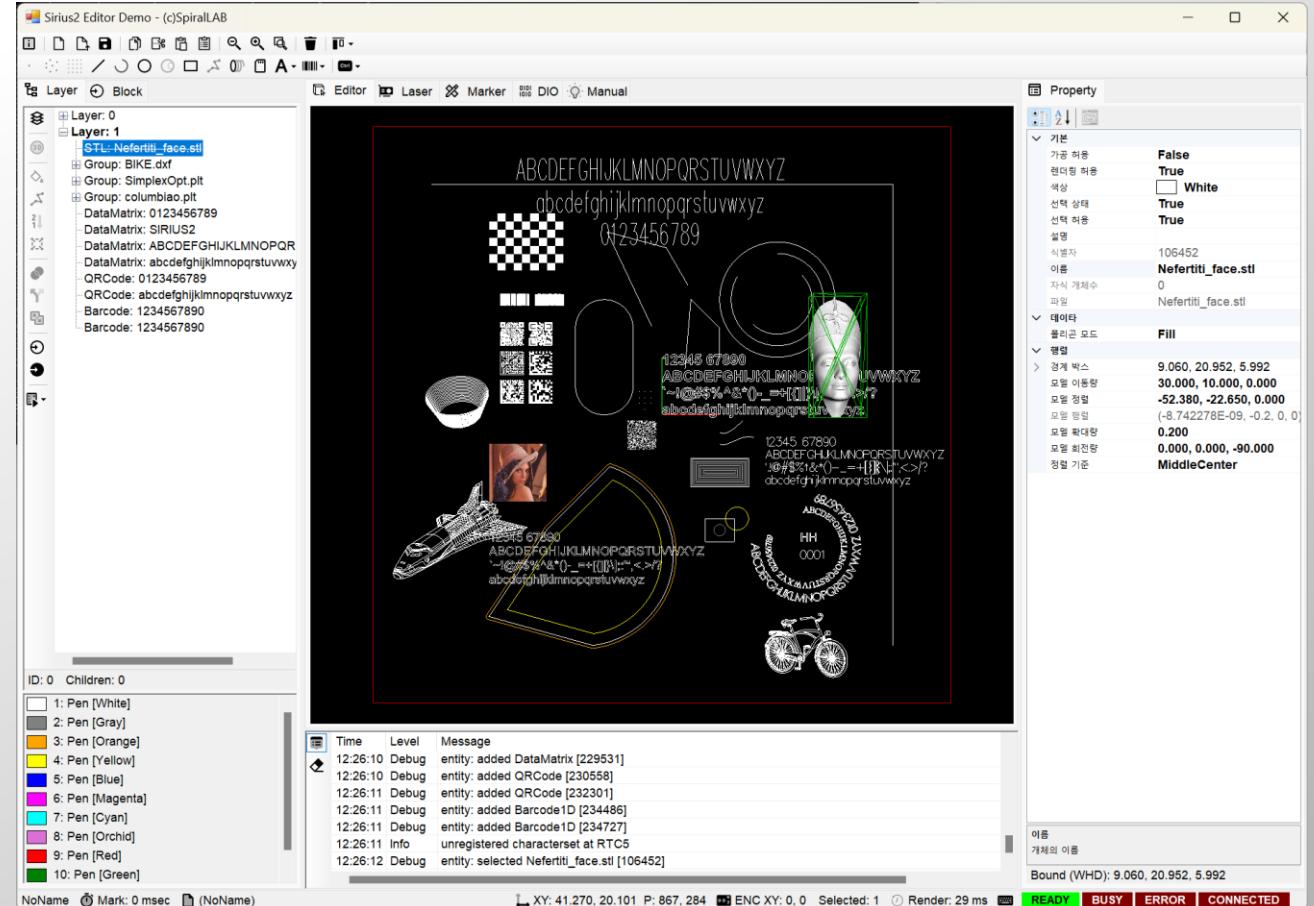
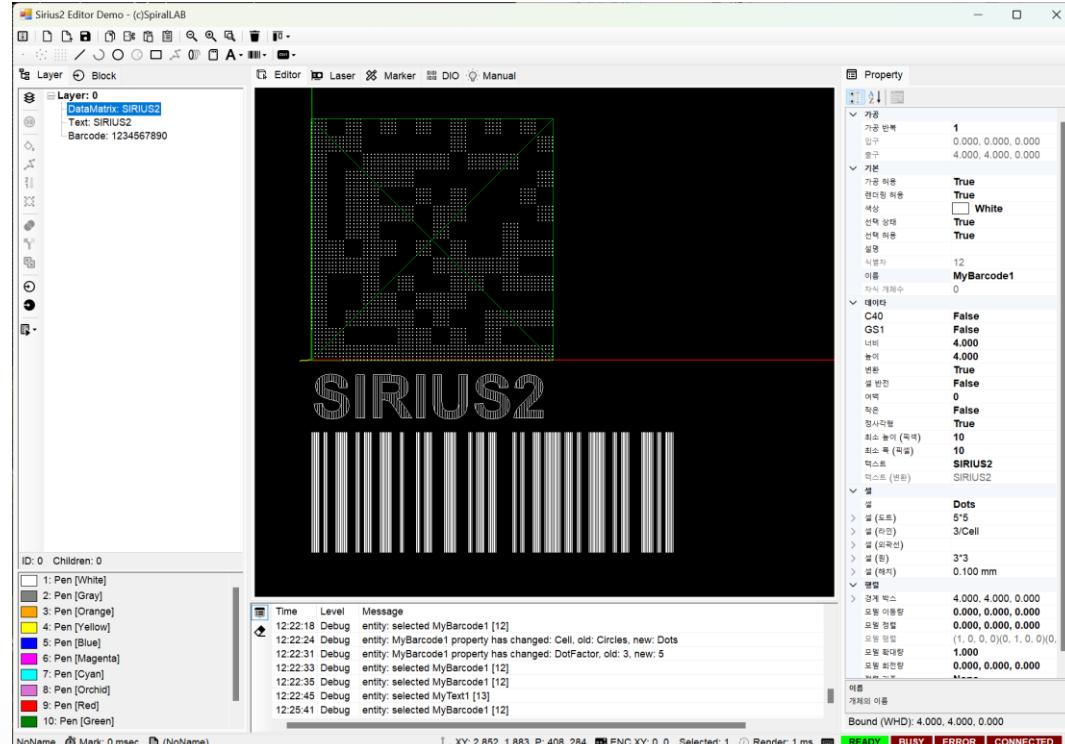
- Remote control by TCP/IP and SERIAL communication
- Change recipe
- Query marker status
- Control marker (start/stop/reset)
- Read/write entity property values
- Read/write script property values

# Integrated XL-SCAN

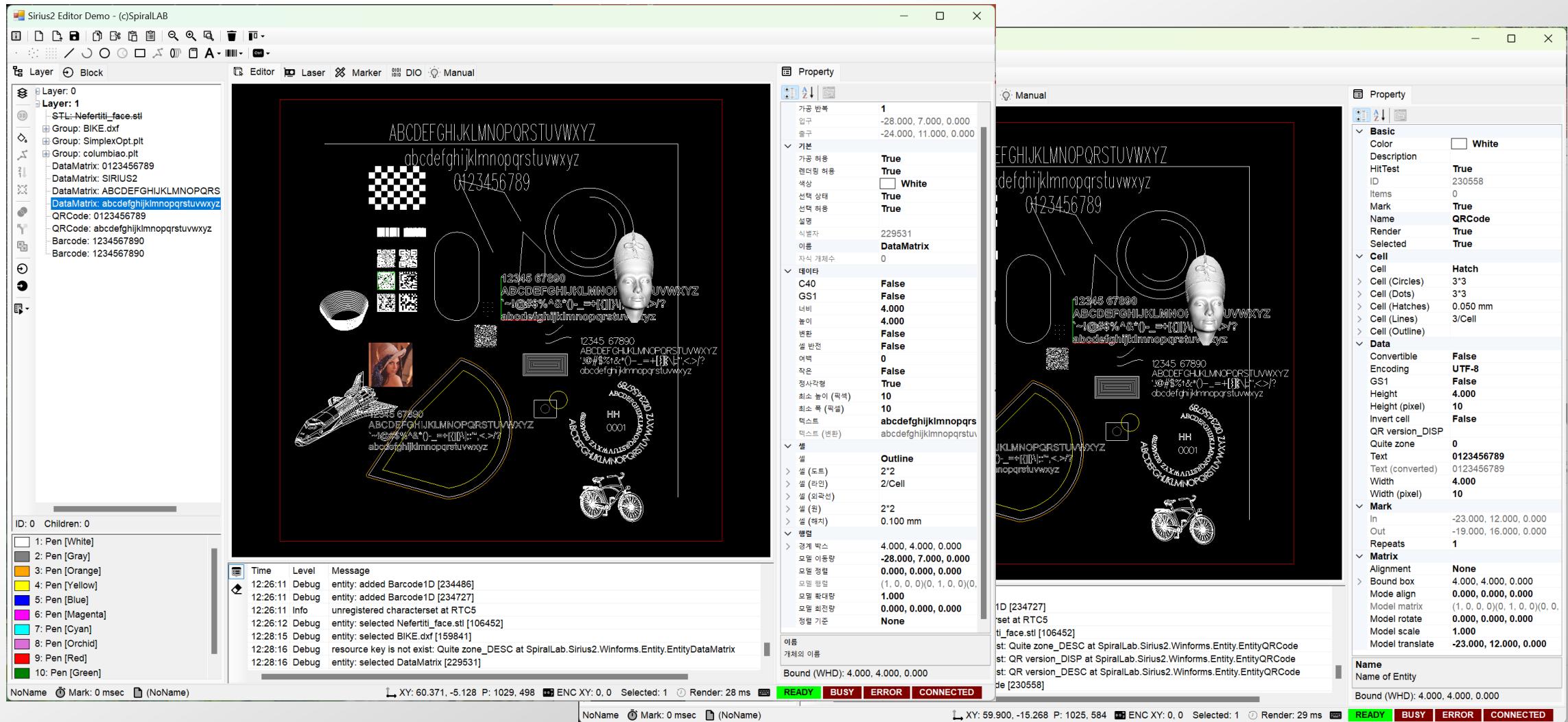


- Easy to change target control platform to XL-SCAN
- No need to modify program(source code)
- Commanded 3D (Z) positions are transformed into Z=0 by automatically
- Executed ‘syncAXIS Viewer’ at simulation mode by automatically

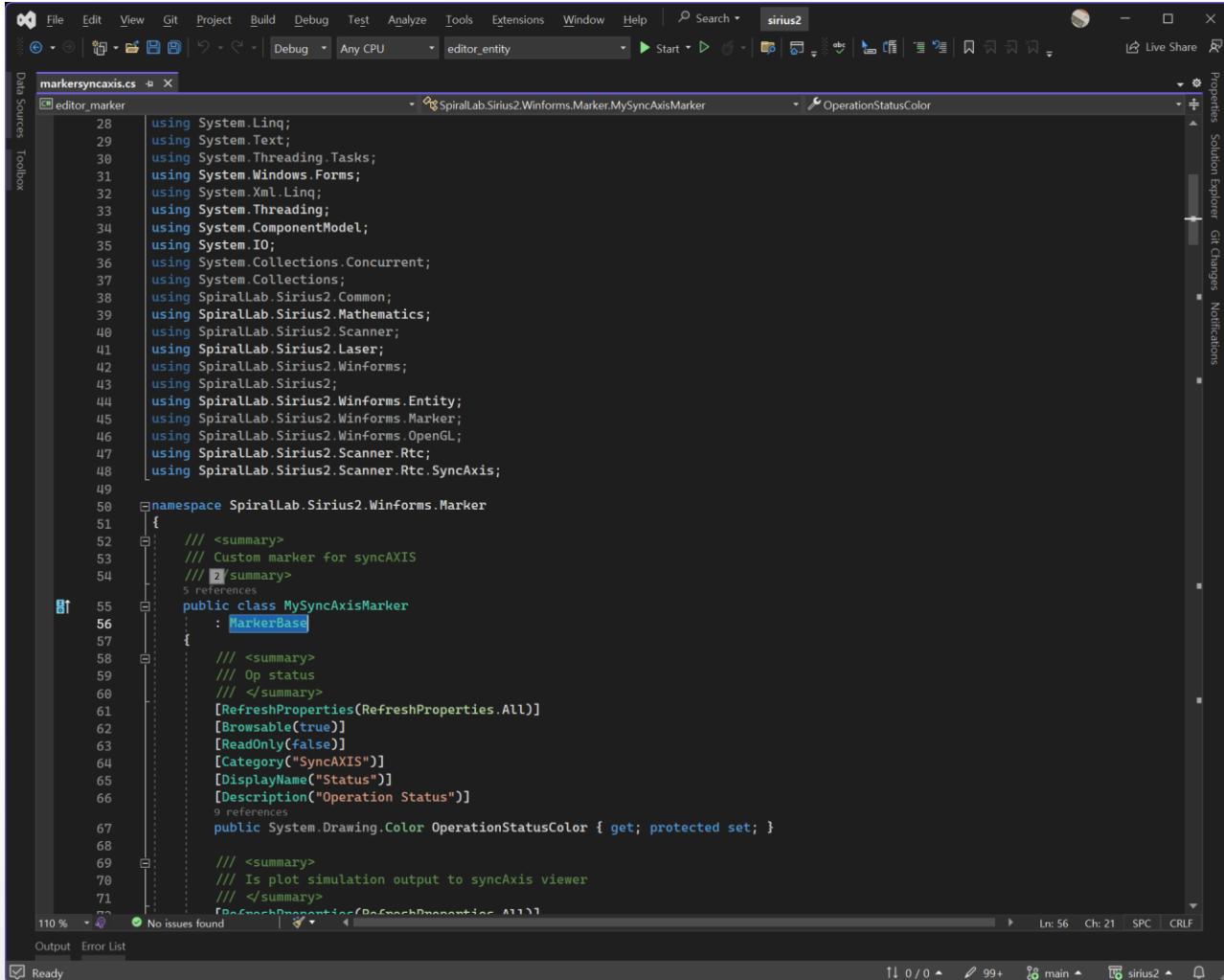
# Powerful and Customizable Demos



# Languages



# Customization



The screenshot shows the Microsoft Visual Studio IDE interface. The title bar says "sirius2". The menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help. The toolbar has icons for search, live share, and other development tools. The code editor window displays a file named "marksyncaxis.cs" with the following content:

```
28 using System.Linq;
29 using System.Text;
30 using System.Threading.Tasks;
31 using System.Windows.Forms;
32 using System.Xml.Linq;
33 using System.Threading;
34 using System.ComponentModel;
35 using System.IO;
36 using System.Collections.Concurrent;
37 using System.Collections;
38 using Spirallab.Sirius2.Common;
39 using Spirallab.Sirius2.Mathematics;
40 using Spirallab.Sirius2.Scanner;
41 using Spirallab.Sirius2.Laser;
42 using Spirallab.Sirius2.WinForms;
43 using Spirallab.Sirius2;
44 using Spirallab.Sirius2.WinForms.Entity;
45 using Spirallab.Sirius2.WinForms.Marker;
46 using Spirallab.Sirius2.WinForms.OpenGL;
47 using Spirallab.Sirius2.Scanner.Rtc;
48 using Spirallab.Sirius2.Scanner.Rtc.SyncAxis;
49
50 namespace SpiralLab.Sirius2.WinForms.Marker
51 {
52     /// <summary>
53     /// Custom marker for syncAXIS
54     /// </summary>
55     public class MySyncAxisMarker
56         : MarkerBase
57     {
58         /// <summary>
59         /// Op status
60         /// </summary>
61         [RefreshProperties(RefreshProperties.All)]
62         [Browsable(true)]
63         [ReadOnly(false)]
64         [Category("SyncAXIS")]
65         [DisplayName("Status")]
66         [Description("Operation Status")]
67         public System.Drawing.Color OperationStatusColor { get; protected set; }
68
69         /// <summary>
70         /// Is plot simulation output to syncAxis viewer
71         /// </summary>
72         [RefreshProperties(RefreshProperties.All)]
73     }
74 }
```

The Solution Explorer, Properties, and Task List panes are visible on the right side of the IDE.

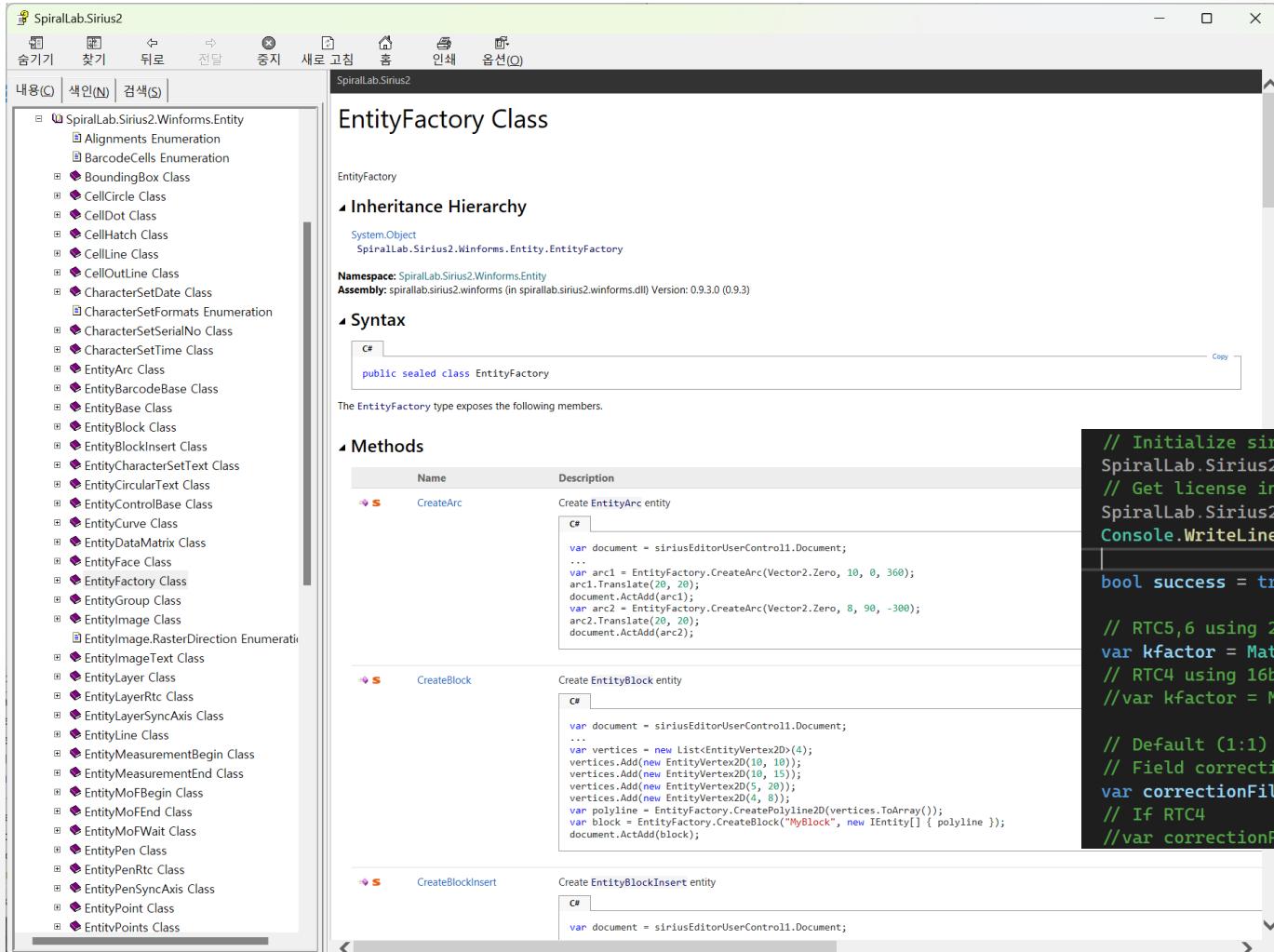
- Customer can be programming own laser application with easily
- Laser (open sourced)
- MarkerRtc (open sourced)
- SiriusEditor (open sourced)

# Repository at GitHub



- Easy to use demo programs
- Included executable editor programs
- Customizable laser, marker and editors
- Repository : <https://github.com/labspiral/sirius2>

# Documentation



The screenshot shows the EntityFactory Class documentation in Visual Studio. The left pane displays a tree view of the Entity namespace, including classes like Aligments Enumeration, BarcodeCells Enumeration, BoundingBox Class, CellCircle Class, CellDot Class, CellHatch Class, CellLine Class, CellOutline Class, CharacterSetDate Class, CharacterSetFormats Enumeration, CharacterSetSerialNo Class, CharacterSetTime Class, EntityArc Class, EntityBarcodeBase Class, EntityBase Class, EntityBlock Class, EntityBlockInsert Class, EntityCharacterSetText Class, EntityCircularText Class, EntityControlBase Class, EntityCurve Class, EntityDataMatrix Class, EntityFace Class, EntityFactory Class, EntityGroup Class, EntityImage Class, EntityImage.RasterDirection Enumeration, EntityImageText Class, EntityLayer Class, EntityLayerRtc Class, EntitySyncAxis Class, EntityLine Class, EntityMeasurementBegin Class, EntityMeasurementEnd Class, EntityMoFBegin Class, EntityMoFEnd Class, EntityMoFWait Class, EntityPen Class, EntityPenRtc Class, EntityPenSyncAxis Class, EntityPoint Class, and EntityPoints Class.

The right pane shows the EntityFactory Class documentation with sections for Inheritance Hierarchy, Syntax, and Methods.

- Inheritance Hierarchy:** System.Object → SpiralLab.Sirius2.Winforms.Entity.EntityFactory
- Namespace:** SpiralLab.Sirius2.Winforms.Entity
- Assembly:** spirallab.sirius2.winforms (in spirallab.sirius2.winforms.dll) Version: 0.9.3.0 (0.9.3)
- Syntax:**

```
C#
public sealed class EntityFactory
```
- Methods:**
  - CreateArc**: Create EntityArc entity  
C#

```
var document = siriusEditorUserControl1.Document;
...
var arc1 = EntityFactory.CreateArc(Vector2.Zero, 10, 0, 360);
arc1.Translate(20, 20);
document.ActAdd(arc1);
var arc2 = EntityFactory.CreateArc(Vector2.Zero, 8, 90, -300);
arc2.Translate(20, 20);
document.ActAdd(arc2);
```
  - CreateBlock**: Create EntityBlock entity  
C#

```
var document = siriusEditorUserControl1.Document;
...
var vertices = new List<EntityVertex2D>(4);
vertices.Add(new EntityVertex2D(10, 10));
vertices.Add(new EntityVertex2D(10, 15));
vertices.Add(new EntityVertex2D(5, 20));
vertices.Add(new EntityVertex2D(4, 8));
var polyline = EntityFactory.CreatePolyline2D(vertices.ToArray());
var block = EntityFactory.CreateBlock("MyBlock", new IEntity[] { polyline });
document.ActAdd(block);
```
  - CreateBlockInsert**: Create EntityBlockInsert entity  
C#

```
var document = siriusEditorUserControl1.Document;
```

**Code Snippet (Right Panel):**

```
// Initialize sirius2 library
SpiralLab.Sirius2.Core.Initialize();
// Get license information
SpiralLab.Sirius2.Core.L
Console.WriteLine($"License key: {licenseKey}");
bool success = true;

// RTC5,6 using 20bits resolution
var kfactor = Math.Pow(2, 5);
// RTC4 using 16bits resolution
//var kfactor = Math.Pow(2, 4);

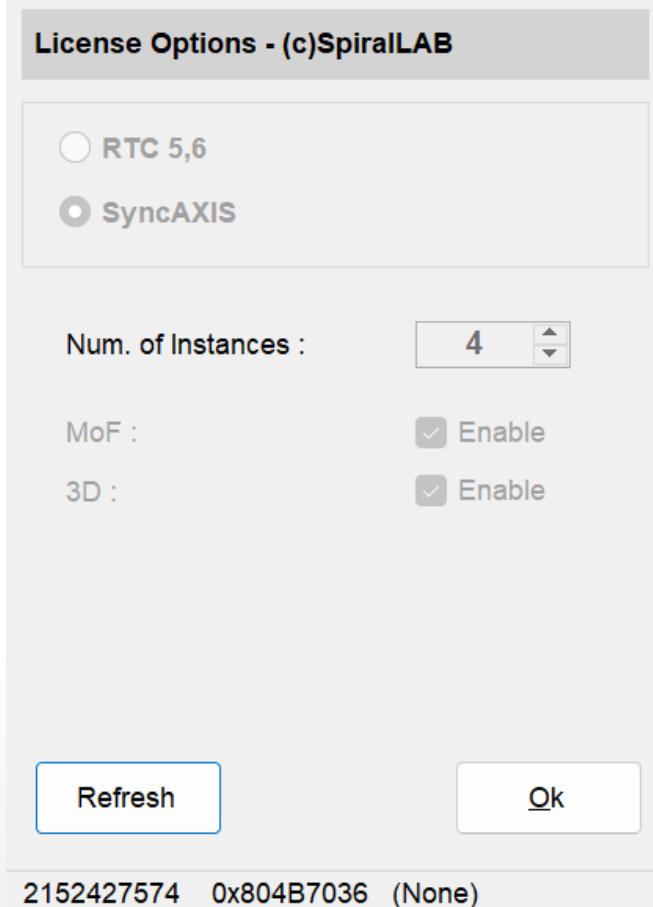
// Default (1:1) correct
// Field correction file
var correctionFile = Path.Combine(Config.LogPath, "correction.txt");
// If RTC4
//var correctionFile = Path.Combine(Config.LogPath, "correction4.txt");

Can be configured minimal log level by Config.LogLevel.
Enabled(or disable) logging feature by Config.IsLogEnable.
Enabled(or disable) log to console by Config.IsLogToConsole.

Returns:
Success of initialization or not
```

- Ready for SDK documentation
- Supported IntelliSense at Visual Studio

# License



- Evaluation copy mode works during 30mins
- Type : RTC 4,5,6 or syncAXIS
- Counts of instances
- Option
  - MoF: Processing on the fly based on encoder
  - 3D : 3D correction by calibration tool