TXL Wizard: Reference Manual

Esteban Marín

2016-05-13

Contents

1 Introduction

This document describes the usage and technical reference of the python program "TXL-Wizard" written by Esteban Marin (estebanmarin@gmx.ch).

1.1 What does it do?

The "TXLWizard" provides routines for generating TXL files (.txl) for the preparation of E-Beam lithography masks using python code. The TXL files can be processed with BEAMER. See the following links:

- `http://genisys-gmbh.com/web/products/beamer.html`
- `http://cad035.psi.ch/LB_index.html`
- `http://cad035.psi.ch/LBDoc/BEAMER_Manual.pdf`

The generated TXL files are also converted to HTML / SVG for presentation in any modern browser or vector graphics application.
Moreover, a command line interface "TXLConverter" provides conversion of existing TXL files to HTML / SVG (See Section **??**).

1.2 Technical Information

The "TXLWizard" is written in python and provides classes in the namespace `EMPyLib.Experiments.Processing.Ebeam.TXLWizard` in the "EMPyLib" package.
The code will run in Python version 2.7+ and 3.1+.
In order to use the "TXLWizard", the "EMPyLib" package must be available as a python package, i.e. either it must be copied to `Path_to_my_python_installation/site-packages/` or to the path where your script is located.
Alternatively, you can also prepend the following command to your python script:

- Windows: `sys.path.append('path to the EMPyLib')`
- Linux: `sys.path.append('path to the EMPyLib')`

2 TXLWizard Example

The following code demonstrates a simple example usage of the "TXLWizard". The resulting image is shown in Figure 1. A more advanced example is shown in Section A

```
1  ######################
2  # Import Libraries #
3  ######################
4
5  # Import TXLWriter, the main class for generating TXL Output
6  import EMPyLib.Experiments.Processing.Ebeam.TXLWizard.TXLWriter
7
8  # Import Pre-Defined Shapes / Structures wrapped in functions
9  import EMPyLib.Experiments.Processing.Ebeam.TXLWizard.
       ShapeLibrary.EndpointDetectionWindows
10 import EMPyLib.Experiments.Processing.Ebeam.TXLWizard.
       ShapeLibrary.Label
11
12 # Import math module for calculations
13 import math
14
15
16 ####################################
17 # Sample / Structure Parameters #
18 ####################################
19
20 # Define all sample parameters
21 SampleParameters = {
22     'Width': 8e3,
23     'Height': 8e3,
24     'Label': 'Simple Demo',
25 }
26
27 # Define all structure parameters
28 StructureParameters = {
29     'Circle': {
30         'Radius': 50,
31         'Layer': 3
32     },
33     'CircleArray': {
34         'Columns': 6,
35         'Rows': 5,
36         'ArrayXOffset': 500,
37         'ArrayYOffset': -500,
38         'ArrayOrigin': [0.75e3, 3e3],
39         'Label': 'R{:d}C{:d}',
40     }
41 }
42
```

```
43
44 ##########################
45 # Initialize TXLWriter #
46 ##########################
47 TXLWriter = EMPyLib.Experiments.Processing.Ebeam.TXLWizard.
       TXLWriter.TXLWriter(
48     Width=SampleParameters['Width'],
49     Height=SampleParameters['Height']
50 )
51
52 ######################
53 # Define Structures #
54 ######################
55
56 ## Sample Label ##
57
58 # Give the sample a nice label
59 SampleLabelObject = EMPyLib.Experiments.Processing.Ebeam.
       TXLWizard.ShapeLibrary.Label.GetLabel(
60     TXLWriter,
61     SampleParameters['Label'],
62     OriginPoint=[0.5e3, 1. * SampleParameters['Height'] / 2. -
           500],
63     FontSize=150,
64     StrokeWidth=20,
65     Layer=1
66 )
67
68
69 ## Endpoint Detection ##
70
71 # Use Pre-Defined Endpoint Detection Windows
72 EMPyLib.Experiments.Processing.Ebeam.TXLWizard.ShapeLibrary.
       EndpointDetectionWindows.GetEndpointDetectionWindows(
73     TXLWriter, Layer=1)
74
75 ## User Structure: Circle ##
76
77 # Create Definition Structure for Circle that will be reused
78 CircleStructure = TXLWriter.AddDefinitionStructure('Circle')
79 CircleStructure.AddPattern('Circle',
80     Center=[0, 0],
81     Radius=StructureParameters['Circle']['Radius'],
82     Layer=StructureParameters['Circle']['Layer']
83
84 )
85
86 # Create array of the definition structure above
```

```
87  CircleArray = TXLWriter.AddContentStructure('CircleArray')
88  CircleArray.AddPattern('Array',
89      ReferencedStructureID=CircleStructure.ID,
90      OriginPoint=StructureParameters['CircleArray']['ArrayOrigin'
            ],
91      PositionDelta1=[
92          StructureParameters['CircleArray']['ArrayXOffset'], 0
93      ],
94      PositionDelta2=[
95          0, StructureParameters['CircleArray']['ArrayYOffset']
96      ],
97      Repetitions1=StructureParameters['CircleArray']['Columns'],
98      Repetitions2=StructureParameters['CircleArray']['Rows']
99  )
100
101
102
103 ############################
104 # Generate Output Files #
105 ############################
106
107 # Note: The suffix (.txl, .html, .svg) will be appended
        automatically
108 TXLWriter.GenerateFiles('Masks/EM160509_Demo_Simple')
```

Content/Example_Simple.py

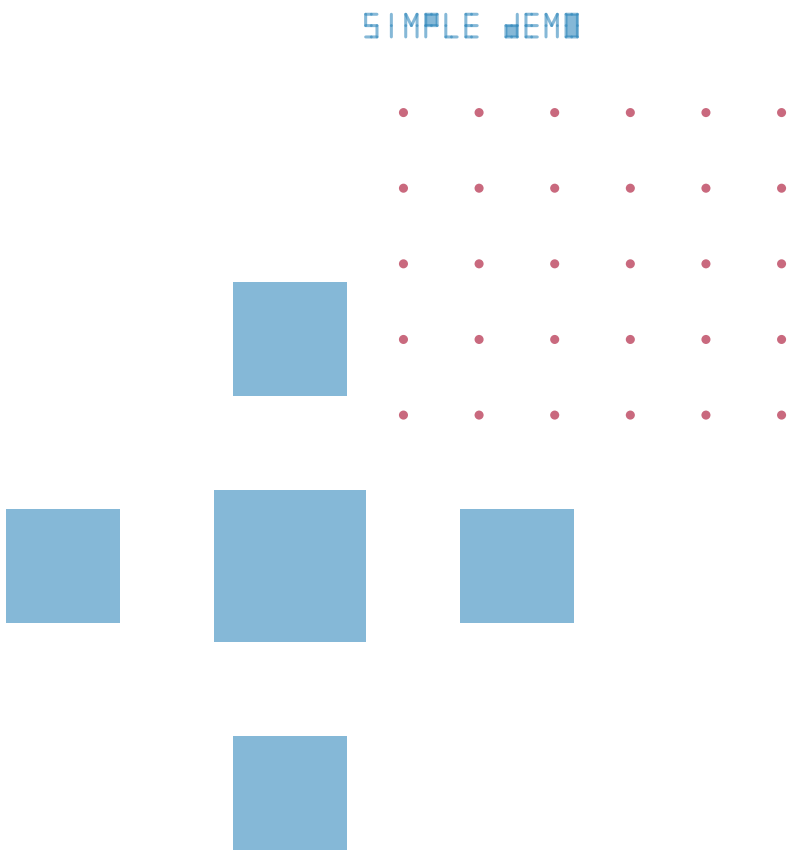

Figure 1: Simple Example: Generated Mask

3 Technische Dokumentation

3.1 Module

3.1.1 Methoden

3.1.2 Eigenschaften

3.2 Strukturen

3.2.1 DataStructure

Die DataStructure ist wie folgt definiert und gilt jeweils für eine Datenbank-Tabelle:

- **Control** `Array` mit Informationen zur Datentabellen-Kontrolle.
  - **Table*** `String` Tabellen-Name
  - **IndexColumn*** `String` Name der Index-Spalte. Standardwert: "ID"
  - **ModifiedTSColumn*** `String` Name der Spalte mit Änderungs-Timestamp. Standardwert: "ModifiedTS"
  - **CategoryColumn*** `String` Name der Kategorien-Spalte. Falls angegeben (muss vom **Type Module** mit **DataCardinality** "1:n" sein), kann nach dieser Spalte kategorisiert werden. (z.B. select-Feld, etc.) Standardwert: "ID"
  - **NoUserInput*** `Bool` Ob kein User-Input in die Daten gelangen darf. Standardwert: `false`
  - **ReadOnly*** `Bool` Ob Datenbank-Tabelle schreibgeschützt ist. Standardwert: `false`
  - **NoPurge*** `Bool` Ob gelöschte Einträge in Datenbank-Tabelle nicht gelöscht werden sollen. Standardwert: `false`
  - **DBConnectionID*** `String` Name der Datenbankverbindung. Standardwert: "ClusterNetAppDB"
- **Columns** `Array` mit Definitionen der einzelnen Daten-Spalten. `Key` ist der Spalten-Name und `Value` ein `Array` mit der Definition:
  - **Type** `String` Typ der Spalte. Kann folgende Werte annehmen:
    - **Integer** Ganzzahl
    - **String** Text
    - **Float** Gleitkommazahl
    - **Checkbox** Checkbox
    - **Date** Datum
    - **Time** Zeit
    - **DateTime** Datum und Uhrzeit
    - **Array** `Array`, wird als JSON gespeichert.
    - **Module** Fremdschlüssel (ID, Referenz) zu einem Eintrag in einem anderen Modul.
    - **FileResource** Dateien

    * **Option** Mehrfachauswahl

  Wird **Type** nicht angegeben, so wird in `$GLOBALS['DefaultFieldStructures']` nach einem Eintrag gesucht. In diesem Fall kann lediglich der **DefaultValue** erhalten bleiben.

- **Validation*** `Array` mit Validierungs-Optionen.
    * **Required*** `Bool` Ob Pflichtfeld. Standardwert: `false`
- **LabelKey*** `String` Key für Language Label. Standardwert: “”
- **TypeSpecificConfiguration*** `Array` spezifische Konfiguration (abhängig von **Type**)

  Type *“Module”*
    * **ModulePath** `String` Pfad des Moduls
    * **MModulePath*** `String` Pfad des M-Moduls der n:m-Relation (benötigt für **DataCardinality** “n:m”)
    * **DataCardinality*** `String` Kardinalität. Erlaubt: “1:n”, “n:m”, Standardwert: “1:n”
    * **NToMTable*** `String` Tabelle für n:m-Relation. (benötigt für **DataCardinality** “n:m”)
    * **NForeignKeyColumn*** `String` Foreign Key Spalte n:m-Relation (n-Tabelle). (benötigt für **DataCardinality** “n:m”)
    * **MForeignKeyColumn*** `String` Foreign Key Spalte für n:m-Relation. (m-Tabelle) (benötigt für **DataCardinality** “n:m”)
    * **ModuleInstance*** `Reference` Referenz zu einer Modul-Instanz. Falls leer, wird diese anhand des **ModulePath** erzeugt.
    * **MModuleInstance*** `Reference` Referenz zu einer M-Modul-Instanz. Falls leer, wird diese anhand des **ModulePath** erzeugt. (für **DataCardinality** “n:m”)
    * **ReplicateToJS*** `Bool` ob Javascript-Modul-Instanz erstellt werden soll. Standardwert: `false`
    * **OverrideInstanceProperties*** `Array` Eigenschaften, die bei Instanziierung des Objekts gesetzt werden sollen. Standardwert: leeres `Array`
    * **OverrideForeignInstanceProperties*** `Array` Eigenschaften, die bei Instanziierung des M-Objekts gesetzt werden sollen. (für **DataCardinality** “n:m”) Standardwert: leeres `Array`
    * **DetailViewType*** `String` Anzeigemodus in der Detailansicht. (für **DataCardinality** “n:m”). Erlaubte Werte: “Full”, “Minimal” Standardwert: “Minimal”
    * **OptionListAdditionalColumns*** `Array` Spalten, die beim Select-option zusätzlich im Frontend vorhanden sein sollen (für **DataCardinality** “1:n”). Standardwert: leeres `Array`
    * **RequirePermission** `Bool` ob relationaler Eintrag gelesen werden können muss, um Datensatz zu lesen. Standardwert: `false`

  Type *“Date”*
    * **AddCurrentTimeToInputDate*** `Bool` ob beim Parsen von User-Input die aktuelle Uhrzeit mitgespeichert werden soll. Standardwert: `false`

***Type** "String"*

* **EncryptDBContent*** `Bool` Ob Wert in der Datenbank verschlüsselt werden soll. Standardwert: `false`
* **RichText*** `Bool` Ob Rich Text editing & anzeige aktiviert ist. Standardwert: `false`
* **AdditionalAllowedHTMLTags*** `String` Weitere erlaubte Tags. Standardwert: ""

***Type** "FileResource"*

* **IsCollection*** `Bool` Ob mehrere Dateien. Standardwert: `false`

***Type** "Integer"*

* **RoundingFactor*** `Float` Rundungsfaktor. Standardwert: 1

***Type** "Float"*

* **RoundingFactor*** `Float` Rundungsfaktor. Standardwert: 1

***Type** "Checkbox"*

* **YesNoLabelKeys*** `Array` spezielle Yes/No Label Keys. Standardwert: leeres `Array`

***Type** "Option"*

* **Options** `Array` Optionen, jede es Element ein `Array` (ID =¿ 'ID', LabelKey =¿ 'LabelKey')

– **Visible*** `Bool` sichtbar? Standardwert: `true`

– **DefaultValue*** `String` Initialwert. Standardwert: **Type**-abhängigen Standardwert gesetzt.

– **Update*** `Bool` ob Feld bei Update in DB geschrieben wird. Standardwert: `true`

– **Insert*** `Bool` ob Feld bei Insert in DB geschrieben wird. Standardwert: `true`

– **NoUserInput*** `Bool` Ob Feld keine Benutzer-Eingaben erhalten darf. Standardwert: `false`

– **DisplayOptions*** Anzeigeoptionen. Wird an Form übergeben. Standardwert: leeres `Array`

* **IsPassword*** `Bool` ob Textfeld als Password dargestellt werden soll. Standardwert: `false`
* **IsCurrency*** `Bool` ob Textfeld als Währung dargestellt werden soll. Bsp: `2'700.35` Standardwert: `false`
* **IsLanguageLabelKey*** `Bool` ob es ein Language Label Key ist. (für **type String**) Standardwert: `false`
* **Multiline*** `Bool` ob mehrzeiliges Textfeld (nur für **String**) Standardwert: `false`
* **FormRowCSSClasses*** `String` zusätzliche Klassen für Formular-Zeile. Standardwert: ""
* **CSSClasses*** `String` zusätzliche Klassen für Formular-Feld. Standardwert: ""

    - **LinkURLs**[*] `Bool` ob URLs im Inhalt gesucht und verlinkt werden sollen (nur für **String**) Standardwert: `false`
    - **DateFormat**[*] `String` Datumsformat für "Date"-Feld. Standardwert: "d.m.Y"
  - **FormFieldParameters**[*] `Array` Parameter, die an das Formular-Objekt übergeben werden. Standardwert: leeres `Array`
    - **IsTabFocusField**[*] `Bool` ob Textfeld im aktuellen Tab standardmässig fokussiert sein soll. Standardwert: `false`
  - **SessionPersistent**[*] `Bool` Ob Wert der Variable Session-persistent gespeichert werden soll. Standardwert: `false`
  - **UserPersistent**[*] `Bool` Ob Wert der Variable User-persistent gespeichert werden soll. Standardwert: `false`

3.2.2 FieldStructure

3.2.3 PageStructure

3.2.4 FormFieldStructure

Appendices

A TXLWizard: Advanced Example

The following code demonstrates the usage of "TXLWizard" in a more advanved way, including labelling of array objects, nested referencing, etc. The generated mask is show in Figure 2.

```
1  ######################
2  # Import Libraries #
3  ######################
4
5  # Import TXLWriter, the main class for generating TXL Output
6  import EMPyLib.Experiments.Processing.Ebeam.TXLWizard.TXLWriter
7
8  # Import Pre-Defined Shapes / Structures wrapped in functions
9  import EMPyLib.Experiments.Processing.Ebeam.TXLWizard.
       ShapeLibrary.EndpointDetectionWindows
10 import EMPyLib.Experiments.Processing.Ebeam.TXLWizard.
       ShapeLibrary.Markers
11 import EMPyLib.Experiments.Processing.Ebeam.TXLWizard.
       ShapeLibrary.Label
12 import EMPyLib.Experiments.Processing.Ebeam.TXLWizard.
       ShapeLibrary.CornerCube
13
14 # Import math module for calculations
15 import math
16
17
18 ####################################
19 # Sample / Structure Parameters #
20 ####################################
21
22 # Define all sample parameters
23 SampleParameters = {
24     'Width': 8e3,
25     'Height': 8e3,
26     'Label': 'GOI Demo CornerCube',
27 }
28
29 # Define all structure parameters
30 StructureParameters = {
31     'CornerCube': {
32         'BridgeLength':8,
33         'ParabolaFocus': 9,
34         'XCutoff': 9,
35         'AirGapX': 3,
36         'AirGapY': 1,
```

```
37          'LabelXOffset': 0,
38          'LabelYOffset': 50,
39          'Label': 'R{:d}C{:d}',
40          'Layer': 2
41      },
42      'Circle': {
43          'Radius': 5,
44          'Layer': 3
45      },
46      'CornerCubeArray': {
47          'Columns': 6,
48          'Rows': 5,
49          'ArrayXOffset': 500,
50          'ArrayYOffset': -500,
51          'ArrayOrigin': [0.75e3, 3e3]
52      }
53  }
54
55
56  ############################
57  # Initialize TXLWriter #
58  ############################
59  TXLWriter = EMPyLib.Experiments.Processing.Ebeam.TXLWizard.
        TXLWriter.TXLWriter(
60      Width=SampleParameters['Width'],
61      Height=SampleParameters['Height']
62  )
63
64  ######################
65  # Define Structures #
66  ######################
67
68  ## Sample Label ##
69
70  # Give the sample a nice label...
71  SampleLabelObject = EMPyLib.Experiments.Processing.Ebeam.
        TXLWizard.ShapeLibrary.Label.GetLabel(
72      TXLWriter,
73      SampleParameters['Label'],
74      OriginPoint=[0.5e3, 1. * SampleParameters['Height'] / 2. -
            500],
75      FontSize=150,
76      StrokeWidth=20,
77      Layer=1
78  )
79  # ...and some other information
80  Alphabet = EMPyLib.Experiments.Processing.Ebeam.TXLWizard.
        ShapeLibrary.Label.GetLabel(
```

```
81      TXLWriter,
82      'abcdefghijklmnopqrstuvwxyz0123456789 megamega ggg ah
            extraaaa rischaaaaar',
83      OriginPoint=[0.5e3, 1. * SampleParameters['Height'] / 2. -
            600],
84      FontSize=50,
85      ShowBar=False,
86      StrokeWidth=3,
87      Layer=1
88  )
89
90  ## Endpoint Detection ##
91
92  # Use Pre-Defined Endpoint Detection Windows
93  EMPyLib.Experiments.Processing.Ebeam.TXLWizard.ShapeLibrary.
        EndpointDetectionWindows.GetEndpointDetectionWindows(
94      TXLWriter, Layer=1)
95
96  ## Alignment Markers ##
97
98  # Use Pre-Defined Alignment Markers
99  EMPyLib.Experiments.Processing.Ebeam.TXLWizard.ShapeLibrary.
        Markers.GetMarkers(
100     TXLWriter, Layer=1)
101
102
103 ## User Structure: Corner Cube ##
104
105 # Create Definition Structure for Corner Cube that will be
        reused
106 CornerCubeDefinition = EMPyLib.Experiments.Processing.Ebeam.
        TXLWizard.ShapeLibrary.CornerCube.GetCornerCube(
107     TXLWriter,
108     ParabolaFocus=StructureParameters['CornerCube']['
            ParabolaFocus'],
109     XCutoff=StructureParameters['CornerCube']['XCutoff'],
110     AirGapX=StructureParameters['CornerCube']['AirGapX'],
111     AirGapY=StructureParameters['CornerCube']['AirGapY'],
112     Layer=StructureParameters['CornerCube']['Layer']
113 )
114
115 # Create Definition Structure for combination of cornercube and
        additional circle
116 FullCornerCubeNoRotation = TXLWriter.AddDefinitionStructure('
        FullCornerCubeNoRotation')
117 FullCornerCubeNoRotation.AddPattern('Reference',
118     ReferencedStructureID=CornerCubeDefinition.ID,
119     OriginPoint=[1. * StructureParameters['CornerCube']['
```

```
        BridgeLength'] / 2., 0]
)
FullCornerCubeNoRotation.AddPattern('Circle',
    Center=[0, 0],
    Radius=StructureParameters['Circle']['Radius'],
    Layer=StructureParameters['Circle']['Layer']
)

# Create definition structure with rotation of entire referenced structure
FullCornerCube = TXLWriter.AddDefinitionStructure('FullCornerCube',
                                                  RotationAngle=45)
FullCornerCube.AddPattern('Reference',
    ReferencedStructureID=FullCornerCubeNoRotation.ID,
    OriginPoint=[0, 0]
)

# Create array of the definition structure above
CornerCubeArrayFine = TXLWriter.AddContentStructure('CornerCubeArrayFine')
CornerCubeArrayFine.AddPattern('Array',
    ReferencedStructureID=FullCornerCube.ID,
    OriginPoint=StructureParameters['CornerCubeArray']['ArrayOrigin'],
    PositionDelta1=[
        StructureParameters['CornerCubeArray']['ArrayXOffset'], 0
    ],
    PositionDelta2=[
        0, StructureParameters['CornerCubeArray']['ArrayYOffset']
    ],
    Repetitions1=StructureParameters['CornerCubeArray']['Columns'],
    Repetitions2=StructureParameters['CornerCubeArray']['Rows']
)


# Add Labels to each array element
for Row in range(1, StructureParameters['CornerCubeArray']['Rows'] + 1):
    for Column in range(1, StructureParameters['CornerCubeArray']['Columns'] + 1):
        RowColumnCountLabel = EMPyLib.Experiments.Processing.Ebeam.TXLWizard.ShapeLibrary.Label.GetLabel(
            TXLWriter,
```

```
156                 StructureParameters['CornerCube']['Label'].format(
                        Row, Column),
157                 OriginPoint=[
158                     StructureParameters['CornerCubeArray']['
                            ArrayOrigin'][0]
159                     + StructureParameters['CornerCubeArray']['
                            ArrayXOffset']
160                     * (Column - 1) + StructureParameters['CornerCube
                            ']['LabelXOffset'],
161                     StructureParameters['CornerCubeArray']['
                            ArrayOrigin'][1]
162                     + StructureParameters['CornerCubeArray']['
                            ArrayYOffset']
163                     * (Row - 1) + StructureParameters['CornerCube']['
                            LabelYOffset']],
164                 FontSize=16,
165                 StrokeWidth=3,
166                 Layer=1,
167                 RotationAngle=45
168             )
169
170
171 ############################
172 # Generate Output Files #
173 ############################
174
175 # Note: The suffix (.txl, .html, .svg) will be appended
        automatically
176 TXLWriter.GenerateFiles('Masks/EM160509_GOI_Demo_CornerCube')
```

Content/Example_GenerateMask.py

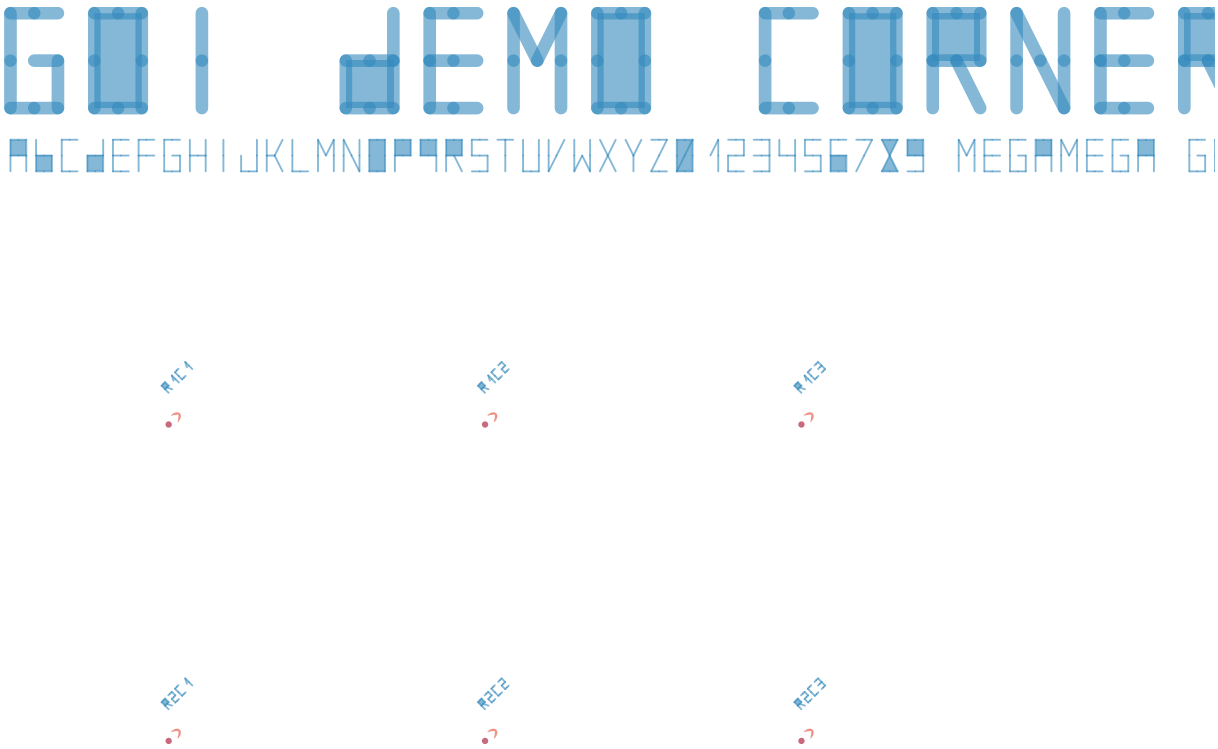

Figure 2: Advanced Example: Part of the Generated Mask