Synexsis Rapid Start UI

To familiarize yourself with development using Synexsis we'll build a quick app to control an Extron matrix switcher.

This control system will run on x32 and ARM devices, so the interface will be designed for UWP or Universal Windows Project. Most of our work for the interface will be done in XAML (which sounds like camel).

Project Setup

Open Visual Studio and create a new Windows UWP project. If you've never created a new project before review our lesson: **Install-VS-Community-Edition**.

We'll be using .NET Standard 2.0 for this project.

A Raspberry Pi and a 7" Raspberry Pi touch screen will be used for this control system. The touch screen size is 800x480. For this system responsive layout is not required. Add a Height property and set it to 480. Add a Width property with a value of 800. Note xlmns:local="using:…." namespace is lighter in color, which indicates it is safe to delete.
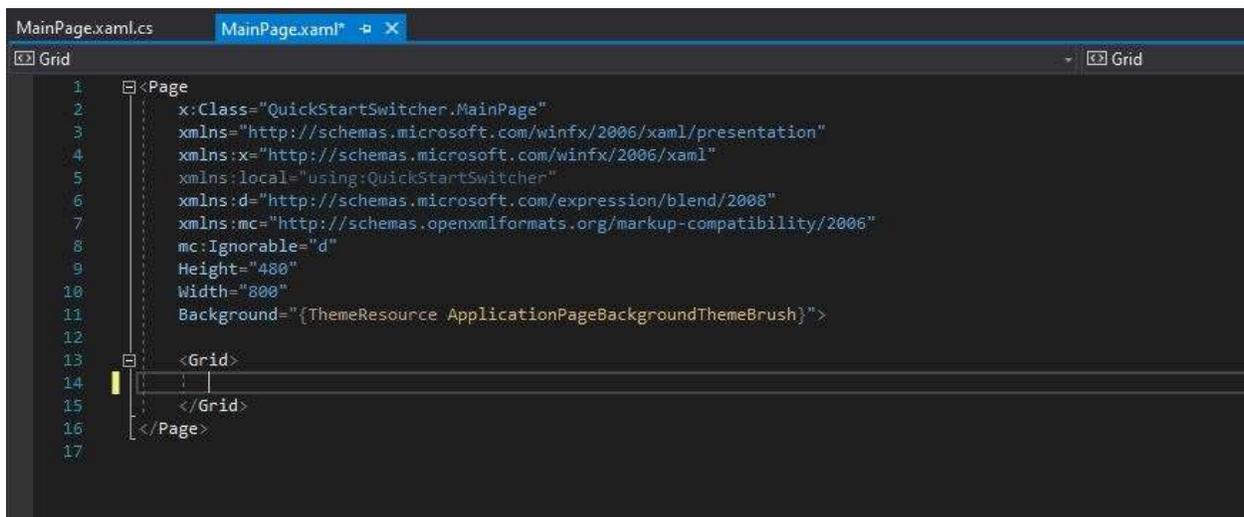
```
MainPage.xaml.cs        MainPage.xaml*  ⊟ ✕
<> Grid                                                          ▼  <> Grid
    1    ⊟<Page
    2        x:Class="QuickStartSwitcher.MainPage"
    3        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    4        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    5        xmlns:local="using:QuickStartSwitcher"
    6        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    7        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    8        mc:Ignorable="d"
    9        Height="480"
   10        Width="800"
   11        Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
   12
   13        <Grid>
   14
   15        </Grid>
   16   </Page>
   17
```

Create a Base Layout

Even though the screen size is fixed, it's still a good idea to use proportional (or star) sizing. Our base design will have text blocks for Input and Output labels, buttons for the switcher channels and a Take/Enter button.

```
<Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="*"/>
        <RowDefinition Height="*"/>
        <RowDefinition Height="*"/>
        <RowDefinition Height="*"/>
        <RowDefinition Height="*"/>
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="*"/>
        <ColumnDefinition Width="*"/>
        <ColumnDefinition Width="*"/>
        <ColumnDefinition Width="*"/>
        <ColumnDefinition Width="*"/>
        <ColumnDefinition Width="*"/>
    </Grid.ColumnDefinitions>
</Grid>
```

Like building a table for a web page, add column and row definitions as show in the code snippet above. Finally, set the Background color to White.

When complete, your design view will look something like this:
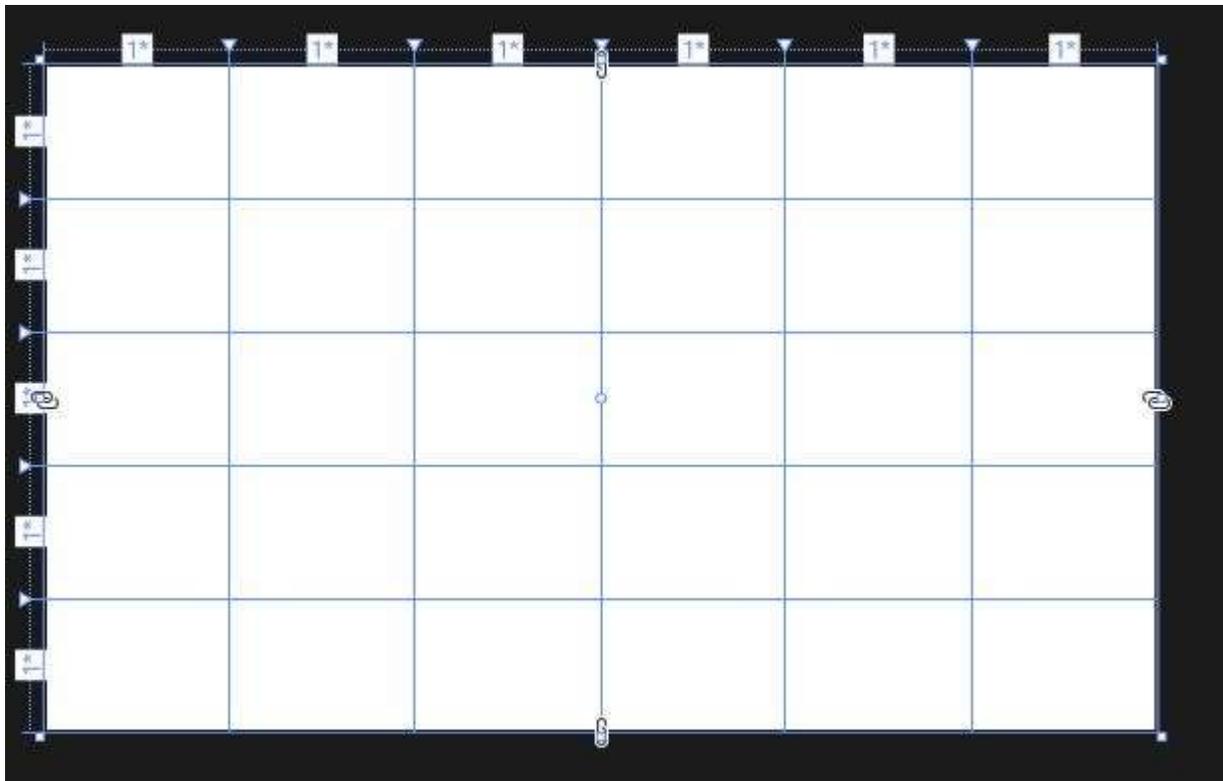
Adding channel buttons

Add the button for Input 1 and name it **BtnInput1**. Place it in the first row and first column with a four pixel margin around all sides. *Note the grid is zero-based starting with the upper-left cell.*

```
39              FontSize="36"
40              Text="Output"/>
41      <Button
42          x:Name="BtnInput1"
43          Grid.Row="1"
44          Grid.Column="1"
45          Margin="4"
46          FontSize="36"
47          HorizontalAlignment="Stretch"
48          VerticalAlignment="Stretch"
49          Content="1"/>
50      </Grid>
51  </Page>
```

The button will fill the cell by setting the horizontal and vertical alignment of the button to "Stretch". For this tutorial the default font will be used and set to 36 pixels. The Content property indicates the text which will appear on the button.

Copy the input 1 button and paste it three times. Then modify the Name and Content properties for inputs 2-4. Modify the Column property to place the button in the correct position.

```
40              Text="Output"/>
41      <Button
42          x:Name="BtnInput1"
43          Grid.Row="1"
44          Grid.Column="1"
45          Margin="4"
46          FontSize="36"
47          HorizontalAlignment="Stretch"
48          VerticalAlignment="Stretch"
49          Content="1"/>
50      <Button
51          x:Name="BtnInput2"
52          Grid.Row="1"
53          Grid.Column="2"
54          Margin="4"
55          FontSize="36"
56          HorizontalAlignment="Stretch"
57          VerticalAlignment="Stretch"
58          Content="2"/>
59      <Button
60          x:Name="BtnInput3"
61          Grid.Row="1"
62          Grid.Column="3"
63          Margin="4"
64          FontSize="36"
65          HorizontalAlignment="Stretch"
66          VerticalAlignment="Stretch"
67          Content="3"/>
68      <Button
69          x:Name="BtnInput4"
70          Grid.Row="1"
71          Grid.Column="4"
72          Margin="4"
73          FontSize="36"
74          HorizontalAlignment="Stretch"
75          VerticalAlignment="Stretch"
76          Content="4"/>
77      </Grid>
78  </Page>
79
```

To add the output buttons, copy and paste the four input buttons. Edit the Row property to 3 to place the buttons in the proper position.

Select all four output buttons, then press Ctrl-H to open the Find/Replace tool. Replace "Input" with "Output" to complete adding the channel buttons. Make sure the dropdown menu is set for Selection rather than Current Document or Entire Solution.
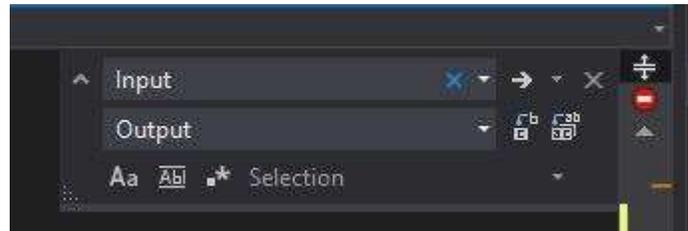
To complete the buttons, add a Take button in Grid.Row 4 and Grid.Column 5.

Adding Text Blocks

Label the switcher input and output rows using a text block, which could also be called a label or caption.

That's it for the user interface. In the next lesson, we'll wire everything up.