



Please note that Cypress is an Infineon Technologies Company.

The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content

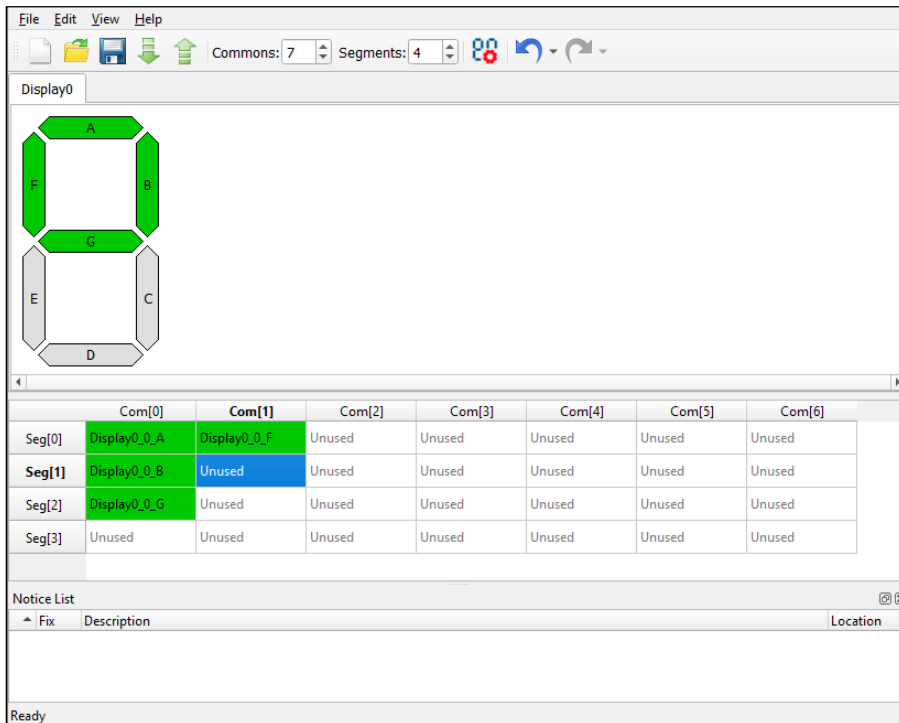
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers

Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

Overview

The Segment LCD (SegLCD) Configurator is a stand-alone tool included with the ModusToolbox software and used to generate display structures for the [SegLCD Driver](#). The tool is supported on Windows, Linux, and macOS.



The SegLCD Configurator supports the following types of displays:

- bar graph (1pixel per symbol)
- 7-segment
- 14-segment
- 16-segment
- 5x8 dot matrix (40 pixels per symbol)

See the [Display Editor](#) section for more information about display types.

Supported Software

Name	Version	Link
Cypress PSoC 6 Peripheral Driver Library	1.3.1 and later	https://github.com/cypresssemiconductorco/psoc6pd

Definitions

- **LCD** – Liquid Crystal Display
- **Glass** – An LCD glass with one or more displays (for example, one 7-segment display and one bar-graph display).
- **Display** – A block of symbols that have the same type on an LCD glass to indicate a multi-digital number or character string.
- **Symbol** – A block of pixels on an LCD glass to indicate a single digit or character.
- **Pixel** – A basic displaying item. This can be a segment of a 7-segment symbol (thus called a "segment"), a pixel of a dot-matrix display, or a stand-alone arbitrarily-shaped display element. Each pixel has a unique set of common and segment lines within one LCD glass.
- **Common line (Com/COM for short)** – A common wire/signal from the PSoC MCU to the LCD glass. In the SegLCD Configurator, this is represented as a column in the [Mapping Table](#).
- **Segment line (Seg/SEG for short)** – A segment wire/signal from the PSoC MCU to the LCD glass. In the SegLCD Configurator, this is represented as a row in the [Mapping Table](#).

Launch the SegLCD Configurator

You can launch the SegLCD Configurator various ways: as a stand-alone tool, from the Device Configurator, from the Eclipse IDE for ModusToolbox or from the command line. Then, you can either use the generated source with an Eclipse IDE application, or use it in any software environment you choose.

As a Stand-Alone Tool

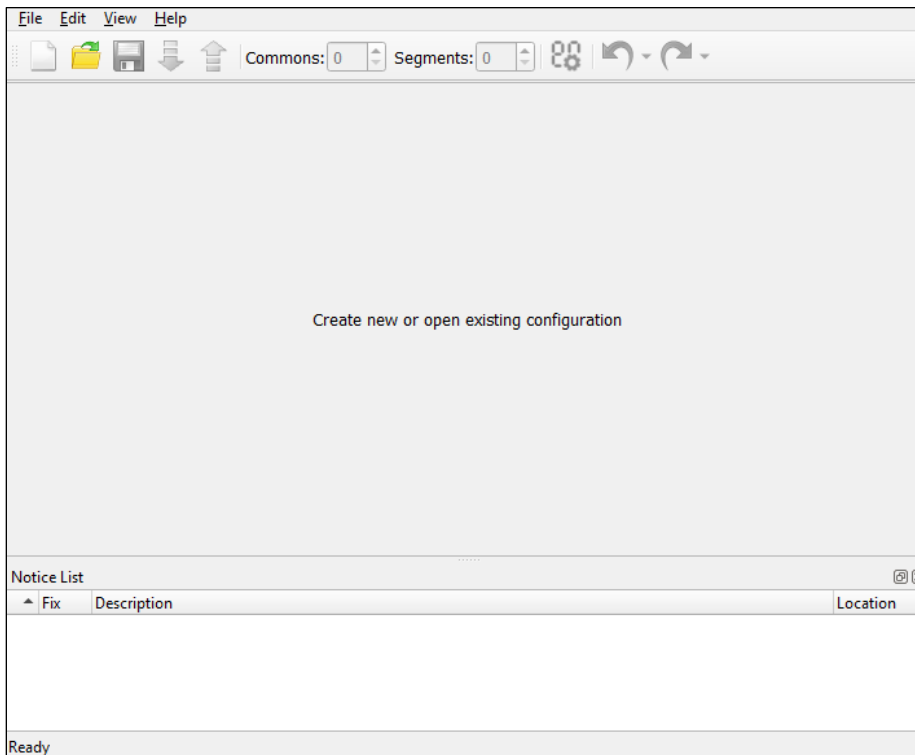
You can launch the SegLCD Configurator as a stand-alone tool. By default, it is installed here:

```
<install_dir>/ModusToolbox/tools_<version>/seglcd-configurator
```

On Windows, you can launch the tool from the **Start** menu.

For other operating systems, navigate to the install location and run the executable.

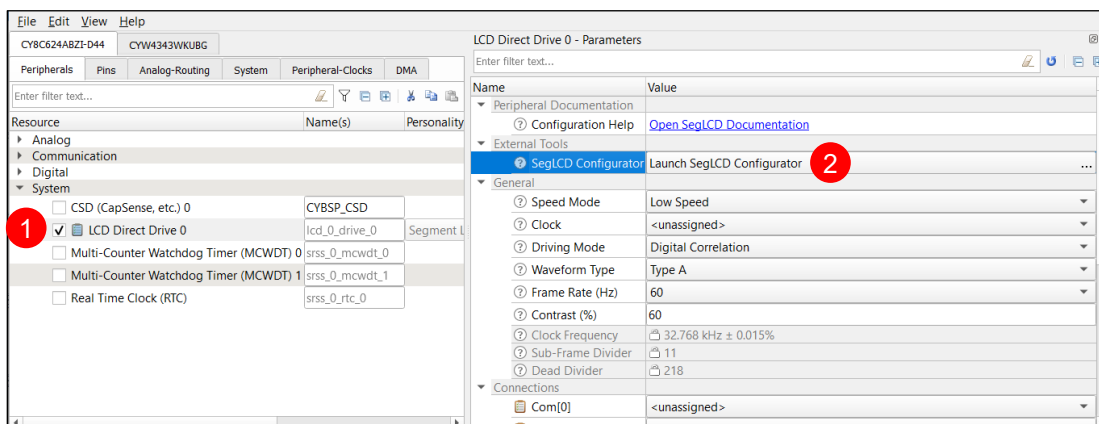
When opened this way, the SegLCD Configurator GUI opens with an untitled configuration file (*.cyseglcd). Save it as a new file and provide a file name, or open another existing *.cyseglcd file.



From the Device Configurator

You can also launch the SegLCD Configurator from the Device Configurator. Refer to the [Device Configurator Guide](#) for more details.

1. On the Device Configurator **Peripherals** tab, enable the **LCD Direct Drive** resource.
2. On the Parameters pane, click the **Launch SegLCD Configurator** button to launch the SegLCD Configurator.



The Device Configurator displays information based on the *design.modus* file and various enabled personalities. When you open the SegLCD Configurator from the Device Configurator, information about the device and the application is passed to the SegLCD Configurator. When you save changes in the SegLCD Configurator, it

updates/generates a *design.cyseglcd* configuration file in the same location as the *design.modus* file. All Board Support Package (BSP) configurators also generate/update firmware in the BSP's "GeneratedSource" folder.

From the Eclipse IDE

If you are using the Eclipse IDE included with ModusToolbox, you can launch the SegLCD Configurator by right-clicking on a project in the Project Explorer and selecting **ModusToolbox > SegLCD Configurator**. Or you can use the IDE Quick Panel. Refer to the [Eclipse IDE for ModusToolbox User Guide](#) for more details.

Launching the SegLCD Configurator from the IDE is similar to opening from the Device Configurator, because the SegLCD configuration file (*design.cyseglcd*) will be related to the *design.modus* file in the BSP.

From the Command Line

You can run the `seglcd-configurator` executable from the command line. However, there are only a few reasons to do this in practice. There is also a `seglcd-configurator-cli` executable which re-generates source code based on the latest configuration settings from a command-line prompt or from within batch files or shell scripts. The exit code for the `seglcd-configurator-cli` executable is zero if the operation is successful, or non-zero if the operation encounters an error. To use the `seglcd-configurator-cli` executable, provide at least the `--config` argument with a path to the configuration file.

For more information about command-line options, run the `seglcd-configurator` or `seglcd-configurator-cli` executable using the `-h` option.

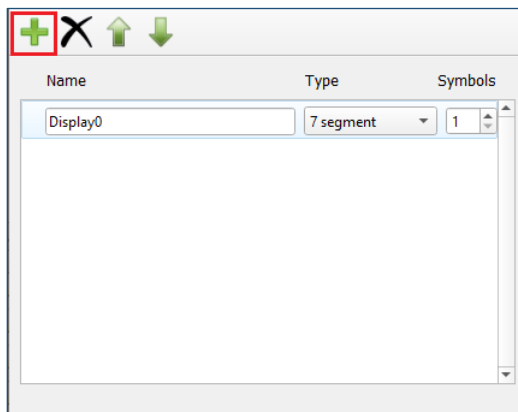
Quick Start

This section provides a simple workflow for how to use the tool.

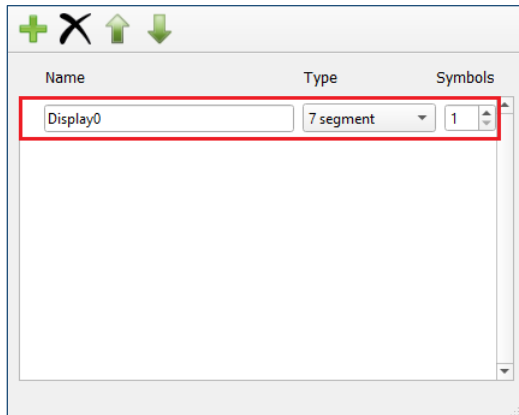
1. [Launch the SegLCD configurator.](#)
2. Click the **Edit Displays** button on the toolbar to open the **Display Editor** dialog.



3. On the Display Editor dialog, click the **Add New Display** button.

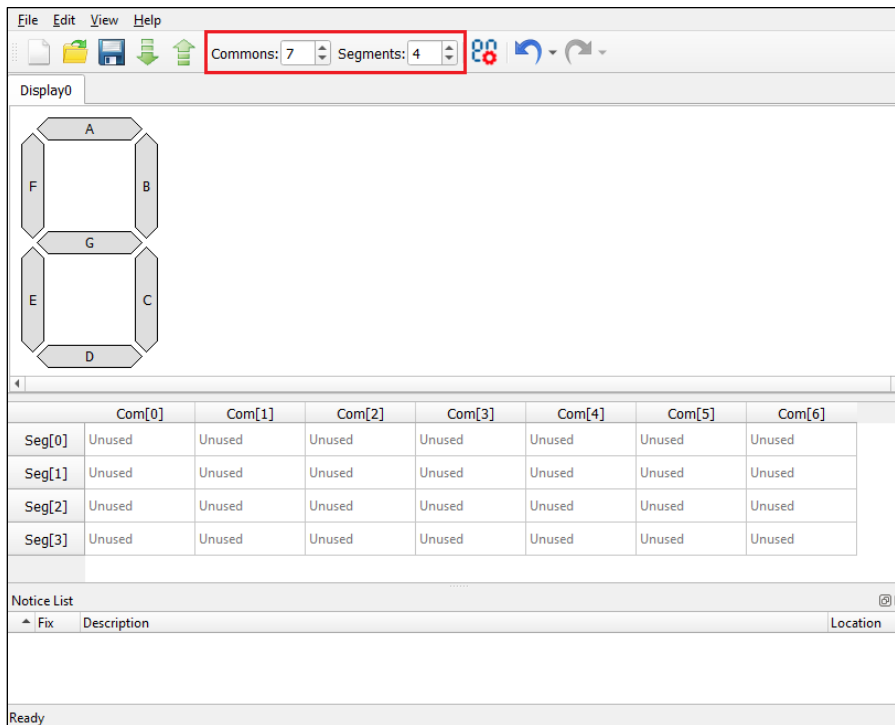


4. A new row appears to specify the display **Name**, select the **Type**, and specify the number of **Symbols**. See the [Display Editor](#) section for more information.

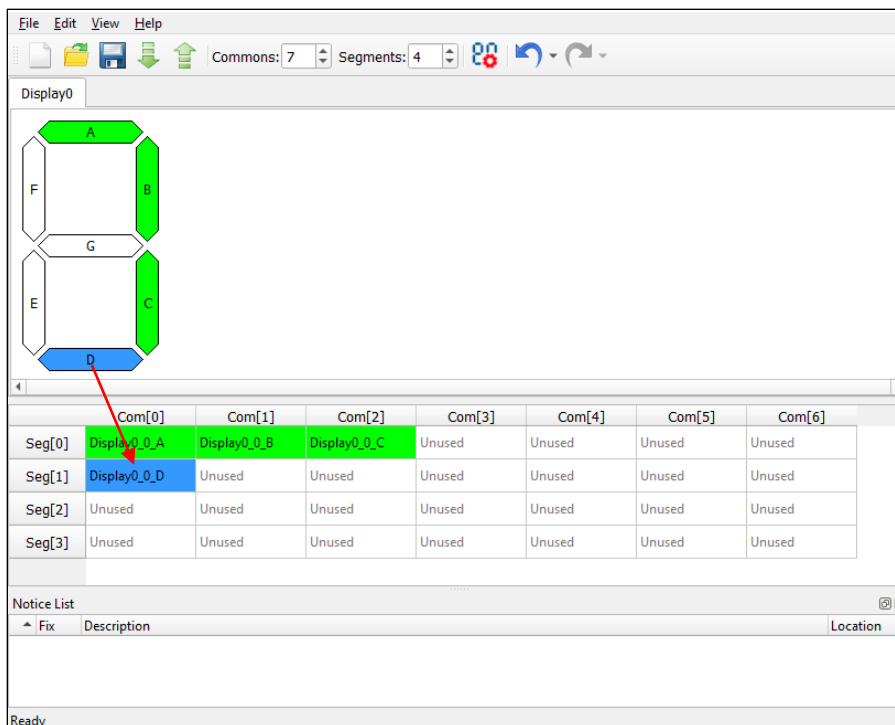


5. Close the dialog by clicking the **X** button or pressing **[Esc]**.

- On the toolbar, specify the number of **Common** and **Segment** LCD connections corresponding to the mapping table dimensions. See the [Mapping Table](#) section for more information.



- On the display view, left-click and hold a pixel of the symbol, drag it onto the mapping table below the display view, and release the mouse button on the desired cell. See the [Mapping Table](#) section for more information.



- Repeat this process for all display pixels.

9. Save the configuration.

See [Code Generation](#).

Code Generation

The SegLCD Configurator generates code in the “GeneratedSource” directory in your BSP, or in the location you specified for a non-Eclipse IDE application. That directory contains the source (.c) and header (.h) files with relevant firmware used by the SegLCD driver. The configuration is stored in the *design.cyseglcd* file next to the *design.modus* file.

After saving the configuration and closing the SegLCD Configurator, complete the configuration:

- If you launched the SegLCD Configurator from the Device Configurator, assign all the unassigned LCD terminals to the GPIO pins in the Device Configurator.
- If you launched the SegLCD Configurator independently, edit the generated code to provide additional LCD terminal definitions for the display array.

GUI Description

Menus

File

- **New** – Creates a new file with new configuration.
- **Open** – Opens and loads an existing file.
- **Save** – Saves the existing file.
- **Save as** – Saves the existing file under a different name.
- **Import** – Imports a specified configuration file.
- **Export** – Exports the current configuration file into a specified file.
- **Exit** – Closes the tool.

Edit

- **Undo** – Undoes the last action or sequence of actions.
- **Redo** – Redoes the last undone action or sequence of undone actions.
- **Edit displays** – Opens the [Display Editor](#) dialog.

View

- **Notice List** – Shows/hides the [Notice List pane](#). The pane is shown by default.
- **Toolbar** – Shows/hides the [toolbar](#).
- **Reset View** – Resets the view to the default.

Help

- **View Help** – Opens this document.
- **About Segment LCD Configurator** – Opens the About box for version information.

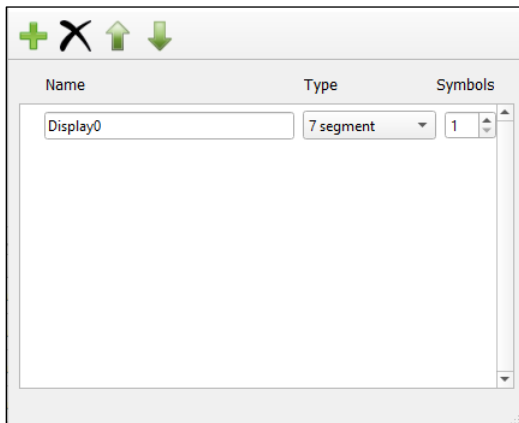
Toolbar

The toolbar contains the same commands as the [File](#) menu. The toolbar also contains:

- **Commons** – Specifies the number of common LCD connections represented as columns on the mapping table.
- **Segments** – Specifies the number of segment LCD connections represented as rows on the mapping table.
- **Edit displays** – Opens the Display Editor dialog.
- **Undo** – Undoes the last action or sequence of actions.
- **Redo** – Redoes the last undone action or sequence of undone actions.

Display Editor

Use this dialog to create and configure displays.



You can open the Display Editor dialog using any of the following ways:




- Click the **Edit** button on the menu.
- Click the **Edit Displays** button on the toolbar
- Keyboard shortcut applicable to the OS.

You can close it multiple ways as well:

- Click the **X** button.
- Keyboard shortcut applicable to the OS.

Display Editor Toolbar

This dialog contains the following toolbar commands:

-  **Add New Display** – Adds a display row to the dialog and a tab to the [Display View](#).
-  **Delete Display** – Deletes the selected display row from the dialog and removes the tab from the Display View.
-  **Move Up/Down** – Moves the selected display row up or down in the dialog, and accordingly rearranges the order of the tabs in the Display View.

Display Row

Use the fields in the display row to enter the display **Name**, select the **Type**, and specify the number of **Symbols**. Use the [Tab] key to switch between the fields, as needed, or select the field using the mouse.

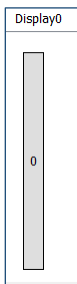
Name

Shows the names of configured displays: Display0, Display1, etc. Each name is unique. It may contain upper or lowercase letters, underscores, and digits. However, the first character cannot be a digit.

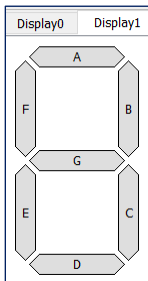
Type

Shows the types of displays to select from the pull-down menu:

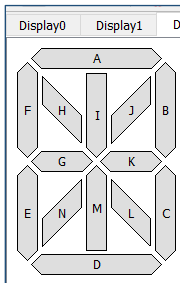
- **Bar/Dial Graph** – Consists of 2 or more pixels. The number of pixels (the bar graph length) is defined by the [Symbols](#) parameter. On the [firmware layer](#), this type is considered as a non-symbolic display with one pixel per symbol. It supports up to 255 segments.



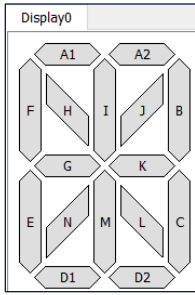
- **7-Segment Display** – Consists of 7 segments per symbol.



- **14-Segment Display** – Consists of 14 segments per symbol.

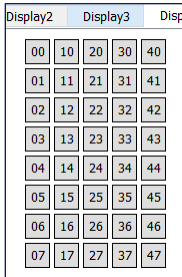


- **16-Segment Display** – Consists of 16 segments per symbol.



Note A decimal point/apostrophe/colon, or any other sign besides the segment display symbol, is not supported as a part of the symbol itself. These signs can be treated as stand-alone pixels.

- **Matrix Display** – For a symbol sized 5 (width) x 8 (height) pixels.

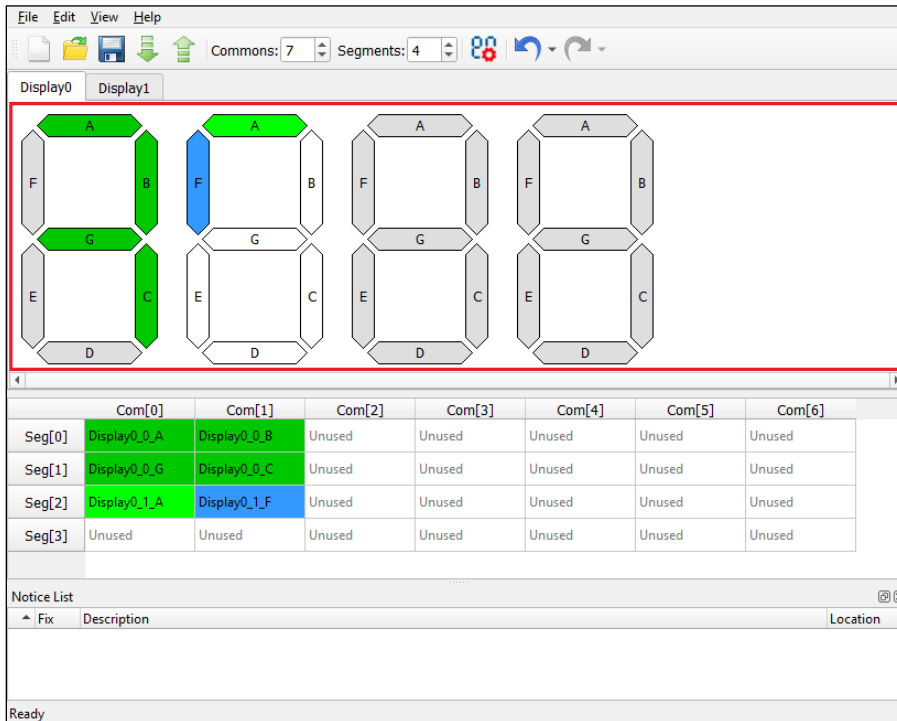


Symbols

The number of display types to include on the corresponding [Display View](#) tab.

Display View

The display view contains one or more tabs that show display types configured in the [Display Editor](#).



A display symbol contains a set of pixels named A, B, C... (or 00, 01, 02 for the matrix display type) by default. The pixel names correspond to cell names in the [Mapping Table](#) (for a pixel placed on the mapping table). The pixels have colors showing the connectivity status:

- **Light Green** – The pixel is connected, and the symbol is currently selected.
- **Dark Green** – The pixel is connected, and the symbol is currently not selected.
- **Blue** – The pixel is currently selected (connected or not).
- **White** – The pixel is not connected, and the symbol is currently selected.
- **Grey** – The pixel is not connected, and the symbol is not currently selected.

Notes

- The same colors are used for pixels in the mapping table, when applicable.
- In this document, symbol segments for displays (for example, 7-segment, 14-segment, etc.), are called “pixels” so that they are not confused with segment LCD connections (physical wires) represented as rows of the mapping table (see [Definitions](#)).

Mapping Table

The mapping table (below the [Display View](#)) is a grid control where columns represent Commons lines, rows represent Segments lines, and cells represent pixels. Each pixel is determined by a unique pair of common and segment lines.

The screenshot shows the ModusToolbox SegLCD Configurator interface. At the top, there are menu options (File, Edit, View, Help) and a toolbar with icons for file operations and configuration. Below the toolbar, there are two tabs: 'Display0' and 'Display1'. The 'Display0' tab is active, showing four display layouts. The first layout shows segments A, B, C, D, E, F, G highlighted in green. The second layout shows segments A, B, C, D, E, F, G highlighted in blue. The third and fourth layouts show segments A, B, C, D, E, F, G in grey. Below the display views is a mapping table with columns for Commons lines (Com[0] to Com[6]) and rows for Segments (Seg[0] to Seg[3]). The mapping table is highlighted with a red border.

	Com[0]	Com[1]	Com[2]	Com[3]	Com[4]	Com[5]	Com[6]
Seg[0]	Display0_0_A	Display0_0_B	Unused	Unused	Unused	Unused	Unused
Seg[1]	Display0_0_G	Display0_0_C	Unused	Unused	Unused	Unused	Unused
Seg[2]	Display0_1_A	Display0_1_F	Unused	Unused	Unused	Unused	Unused
Seg[3]	Unused	Unused	Unused	Unused	Unused	Unused	Unused

Below the mapping table is a 'Notice List' section with columns for 'Fix', 'Description', and 'Location'. The status bar at the bottom indicates 'Ready'.

Connect/Disconnect Display Pixel

To connect a display pixel to a specific Common and Segment, drag a pixel from the [Display View](#) to the mapping table.

The screenshot shows a close-up of the mapping table and a portion of the display view. A red arrow points from a pixel in the display view (segment D, common C) to the mapping table cell for Seg[0], Com[0], which contains 'Display0_0_D'. Another red arrow points from the mapping table cell for Seg[0], Com[1], which contains 'Display0_0_C', to a pixel in the display view (segment C, common A). The mapping table is highlighted with a red border.

	Com[0]	Com[1]	Com[2]
Seg[0]	Display0_0_D	Display0_0_C	Unused
Seg[1]	Unused	Unused	Unused

You can also right-click on the cell to access the context menu and select the desired display, symbol, and pixel.

	Com[0]	Com[1]	Com[2]	Com[3]
Seg[0]	Display0	Symbol #0		
Seg[1]	Display1	Used	Unused	
Seg[2]	Display0_0_F	Display0_0_G	Unused	
Seg[3]	Unused	Unused	Unused	

Display0_0_A
 Display0_0_B
 Display0_0_C
 Display0_0_D
 Display0_0_E
 Display0_0_F
 Display0_0_G

To disconnect a display pixel, select a pixel in the table and drag the selected pixel outside the mapping table. You can also simply press **Delete**.

Change Pixel Name

When a new file opens on the SegLCD GUI, all the cells in the mapping table are labelled as “Unused”.

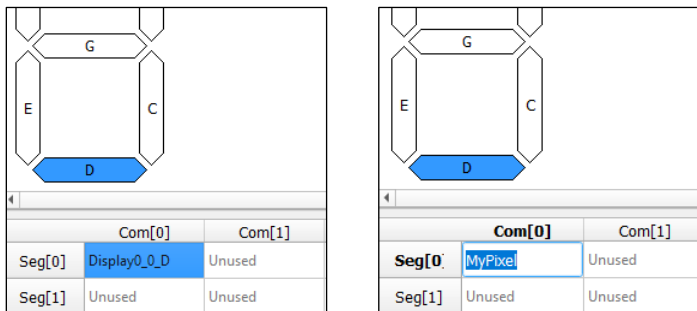
If the user names an unconnected pixel from the mapping table, a define is added to the generated header file. For example,

```
#define <Peripheral instance name>_<User defined pixel name> (CY_SEGLCD_PIXEL( <Common number>UL, <Segment number>UL))
```

By default, display pixels connected to Commons and Segments in the mapping table have the following name format:

<display name>_<symbol number>_<pixel name>

To rename a pixel, double-click it in the mapping table to select it, and then replace the text. Each pixel name must be unique. The name can contain uppercase and lowercase letters, underscores, and digits. However, the first character cannot be a digit.



Once the name has been changed, only the first two characters of the pixel name show in the display name.

To return a pixel name to its default value, either delete the name and press **[Enter]** or select the name and type “Unused”.

Notice List

The Notice List pane combines notices (errors, warnings, tasks, and notes) from many places in the configuration into a centralized list. If a notice shows a location, double-click the entry to show the error or warning.

Fix	Description	Location
✖	Only alphanumeric symbols and _ (underscore) are allowed. Must not begin with a digit.	

For more information, refer to the [Device Configurator Guide](#).

References

Refer to the following documents for more information, as needed:

- [Device Configurator Guide](#)
- [Eclipse IDE for ModusToolbox User Guide](#)
- [PDL Reference Guide](#)
- Device Datasheets
- Device Technical Reference Manuals

Version Changes

This section lists and describes the changes for each version of this tool.

Version	Change Descriptions
1.0	New tool.
1.1	Added the Undo / Redo feature.
1.20	Updated versioning to support patches.
	Added Copy feature to the Notice List.
	Changed unused pixels' names from "PIX{N _e }" to "Unused" in the mapping table.
	Fixed the colors in DarkMode.
	Added the correct error location in the Notice List.
1.21	Updated versioning to support the updated backend.

© Cypress Semiconductor Corporation, 2019-2020. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, ModusToolbox, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.