ModusToolbox™ Segment LCD Configurator user guide

ModusToolbox™ tools package version 3.6.0
Segment LCD Configurator version 1.90.0

About this document

A newer version of this document may be available on the web here.

Scope and purpose

The Segment LCD Configurator is used to generate display structures for the Segment LCD Driver.

Intended audience

This document helps application developers understand how to use the Segment LCD Configurator as part of creating a ModusToolbox™ application.

Document conventions

Convention	Explanation
Bold	Emphasizes heading levels, column headings, menus and sub-menus
Italics	Denotes file names and paths.
`Monospace`	Denotes APIs, functions, interrupt handlers, events, data types, error handlers, file/folder names, directories, command line inputs, code snippets
File > New	Indicates that a cascading sub-menu opens when you select a menu item

Abbreviations and definitions

The following define the abbreviations and terms used in this document:

- Application – One or more projects related to each other
- Configurator – A GUI-based tool used to configure a resource
- 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; 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 Segment LCD 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 represented as a row in the Mapping table.

About this document

Reference documents

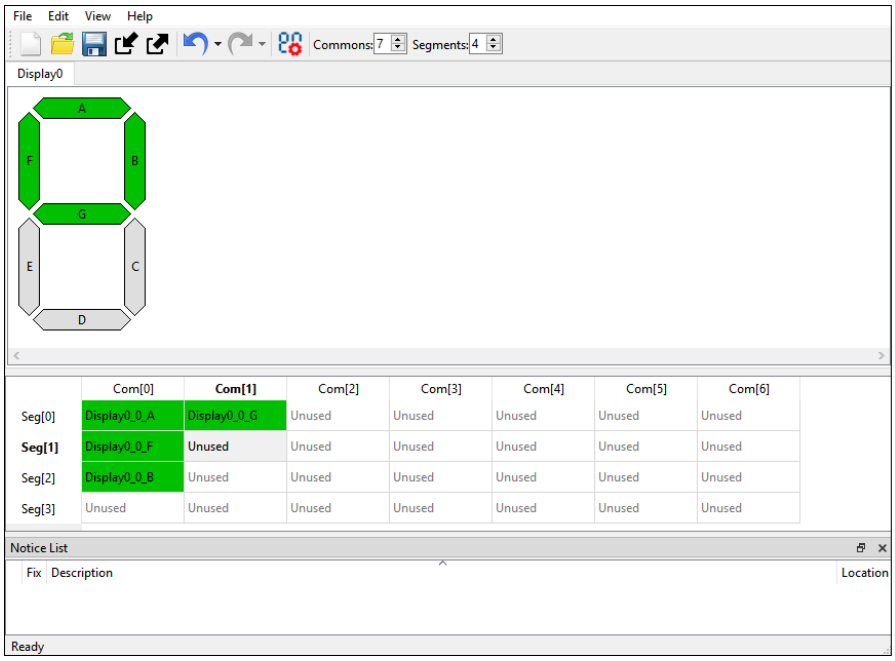
Refer to the following documents for more information as needed:

- Device Configurator user guide
- Eclipse IDE for ModusToolbox™ user guide
- PSoC™ 6 PDL
- MTB CAT1 PDL
- MTB CAT2 PDL
- Device datasheets
- Device technical reference manuals

Table of contents

1 Overview

The Segment LCD Configurator is a stand-alone tool included with the ModusToolbox™ software and used to generate display structures for the Segment LCD Driver. Section "Supported software" provides details on the PDL libraries. The tool is supported on Windows, Linux, and macOS.



The Segment LCD Configurator supports the following types of displays:

- bar graph (1 pixel 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.

1.1 Supported software

Name	Version	Link
PSoC™ 6 Peripheral Driver Library	1.3.1 and later	https://github.com/Infineon/psoc6pdl
MTB CAT1 Peripheral Driver Library	2.0.0 and later	https://github.com/Infineon/mtb-pdl-cat1
MTB CAT2 Peripheral Driver Library	1.3.0 and later	https://github.com/Infineon/mtb-pdl-cat2

2 Launch the Segment LCD Configurator

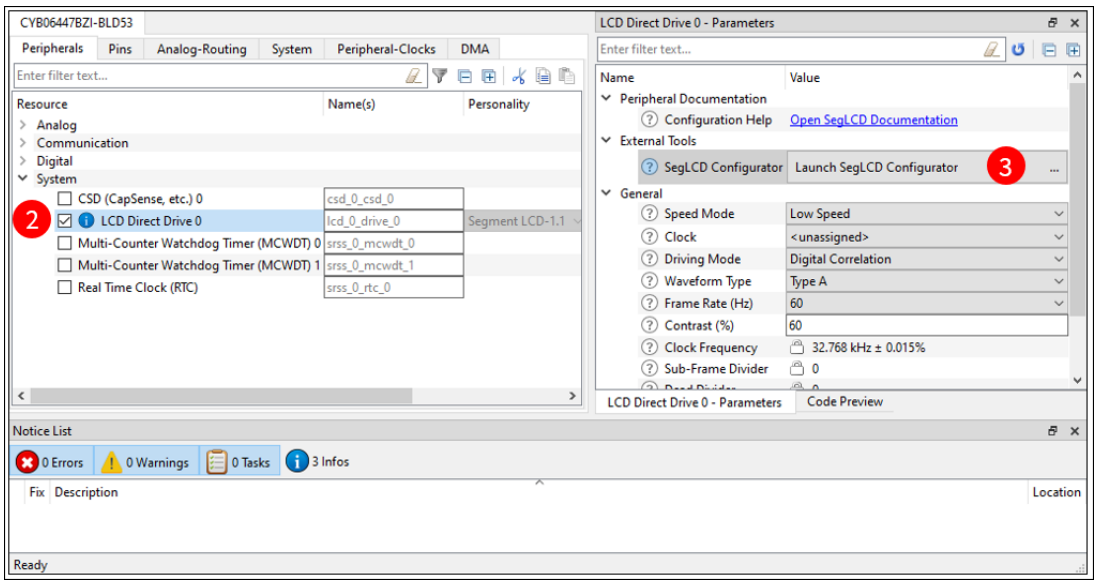
There are several ways to launch the Segment LCD Configurator, as described in this section. However, the best practice is to launch it using the Device Configurator to ensure that your application code remains in sync.

***Note**: The Segment LCD Configurator requires the LCD Direct Drive resource, which you enable using the Device Configurator.*

2.1 From the Device Configurator

1. Open the Device Configurator using one of the methods described in the Device Configurator user guide.
2. On the Peripherals tab, enable the LCD Direct Drive resource if not already enabled.
3. On the Parameters pane, click the Launch Segment LCD Configurator button.

***Note**: You may be asked to save changes.*



After you have created Segment LCD configuration for your application and used the Device Configurator to configure the Common and Segment signals, you can then use the Segment LCD Configurator without the Device Configurator on subsequent updates.

***Note**: If you change the number of Common or Segment signals while using the Segment LCD Configurator, then you must also use the Device Configurator again to reconfigure those signals.*

2.2 make command

As described in the tools package user guide "ModusToolbox™ build system" chapter, you can run numerous make commands in the application directory, such as launching the Segment LCD Configurator. After you have created a ModusToolbox™ application, navigate to the application directory and type the following command in the appropriate bash terminal window:

```
make seglcd-configurator
```

This command opens the Segment LCD Configurator GUI for the specific application in which you are working.

2.3 VS Code and Eclipse IDE

VS Code and Eclipse have tools to launch the SegLCD Configurator from within an open application. Refer to the applicable user guide for more details:

- VS Code for ModusToolbox™ user guide
- Eclipse for ModusToolbox™ user guide

2.4 Executable (GUI)

You can launch the Segment LCD Configurator GUI by running its executable as appropriate for your operating system (for example, double-click it or select it using the Windows **Start** menu). By default, the configurator is installed here:

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

When launched this way, the Segment LCD 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.

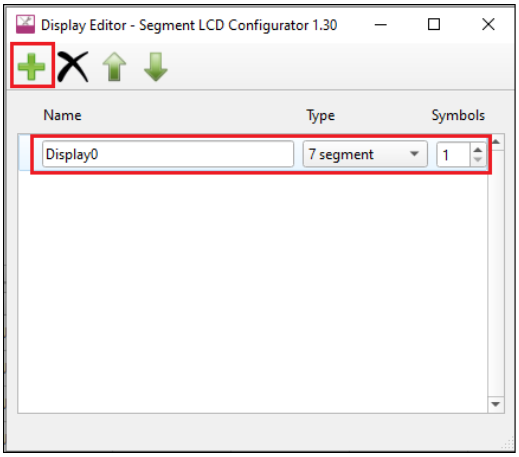
2.5 Executable (CLI)

The Segment LCD Configurator executable can be run from the command line and it has a "cli" version of the executable as well. Running the executable from the command line can be useful as part of batch files or shell scripts to re-generate the source code based on the latest configuration settings. The exit code for the executable is zero if the operation is successful, or non-zero if the operation encounters an error. For more information about the command-line options, run the executable using the `-h` option.

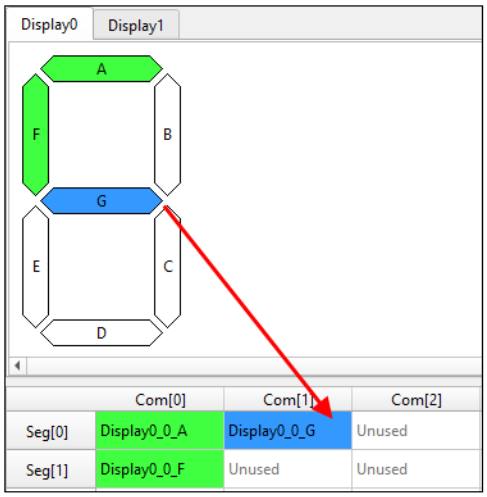
3 Quick start

This section provides a simple workflow for how to use the Segment LCD Configurator.

1. Launch the Segment LCD Configurator from the Device Configurator.
2. On the Segment LCD Configurator toolbar, click the **Edit Displays** button to open the Display Editor dialog.
3. On the Display Editor dialog toolbar, click the **Add New Display** button. In the new row, select the **Type** and specify the display **Name** and number of **Symbols**.

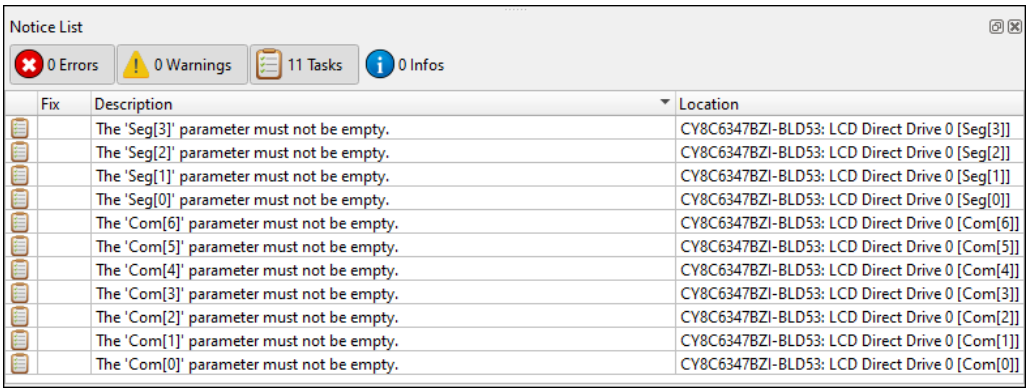


4. Close the dialog by clicking the **X** button or pressing [**Esc**].
5. On the Segment LCD Configurator toolbar, specify the number of **Common** and **Segment** LCD connections corresponding to the mapping table dimensions. See the Mapping table section for more information.
6. On the Display view, 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.

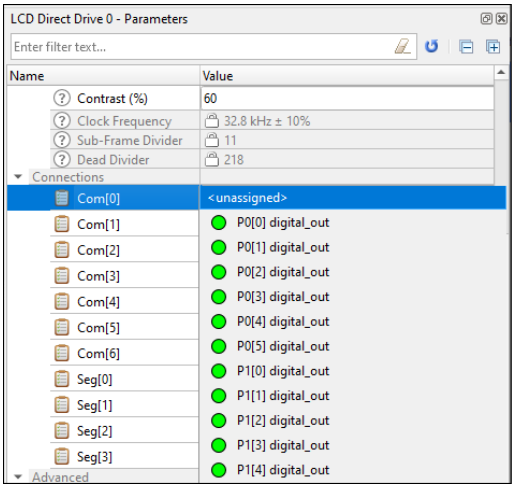


7. Repeat this process for all display pixels.
8. Save the configuration and close the Segment LCD Configurator.

   Back on the Device Configurator, there are several tasks in the Notice List, which correlate to the number of Common and Segment signals you configured.



9. Double-click one of the task icons to jump to the Parameters pane for the corresponding connection parameter. Click the pull-down menu and select the appropriate signal.



10. Repeat the process for every task.

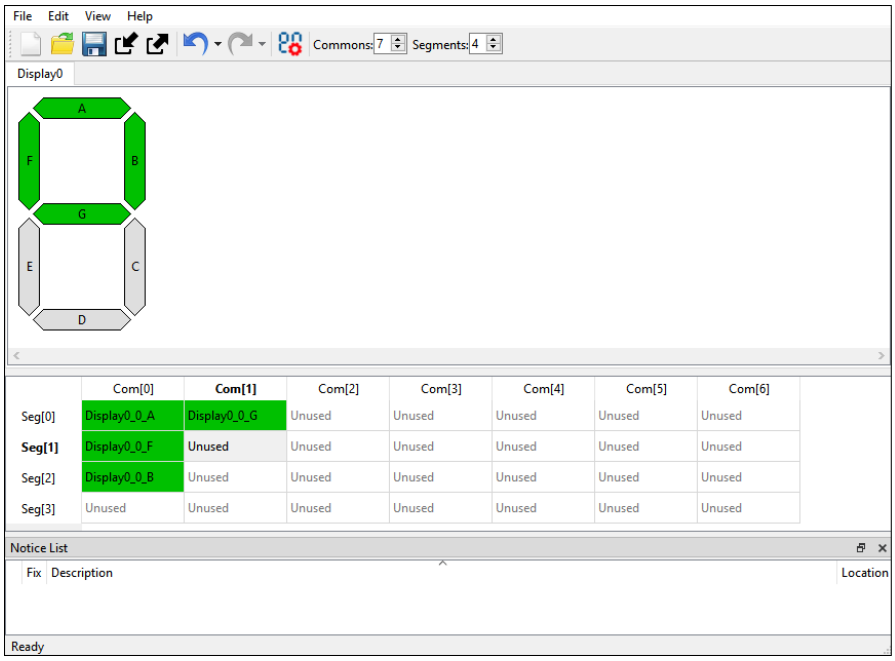
11. When finished, save and close the Device Configurator; see Code generation .

4 Code generation

The Device Configurator displays information based on the *design.modus* file and various enabled personalities. When you open the Segment LCD Configurator from the Device Configurator, information about the device and the application is passed to the Segment LCD Configurator. When you save changes in the Segment LCD Configurator, it updates/generates a *design.cyseglcd* configuration file in the same location as the *design.modus* file, and it passes information back to the Device Configurator.

Saving the configuration files generates code in the *GeneratedSource* subdirectory, which is located next to your configuration files. That subdirectory contains the source (.c) and header (.h) files with relevant firmware used by the Segment LCD driver (refer to section Supported software).

5 GUI description



The Segment LCD Configurator GUI contains menus and display view to configure display tabs, and a Notice List to provide indications.

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.
- **Open in System Explorer** – Opens your computer's file explorer tool to the folder that contains the **.modus* file.
- **Import...** – Imports a specified configuration file.
- **Export...** – Exports the current configuration file into a specified file.
- **Recent Files** – Shows recent files that you can open directly.
- **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

- **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.

Help

- **View Help** – Opens this document.
- **About Segment LCD Configurator** – Opens the About box for version information, with links to open https://www.infineon.com and the current session log file.

5.1 Toolbar

The toolbar provides the basic buttons from the menus to create, open, edit, and save files.

Also, the toolbar contains buttons to configure and edit displays:

- **Edit Displays** – Opens the Display Editor dialog.
- **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.

5.2 Display Editor

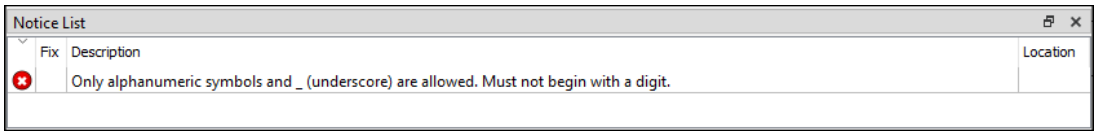
See Display Editor.

5.3 Display View

See Display view.

5.4 Notice List

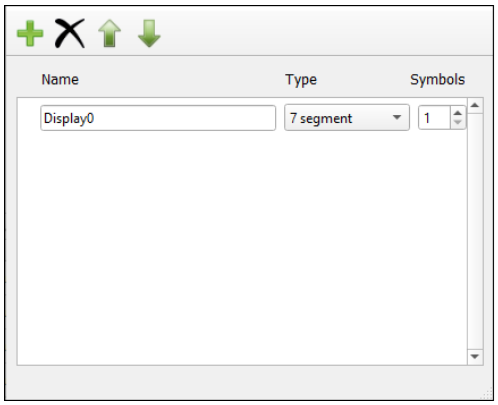
The Notice List pane combines notices (errors, warnings, tasks, and notes) from many places in the configuration into a centralized list. If a notice shows a location, double-click the entry to show the error or warning.



For more information, refer to the Device Configurator guide.

6 Display Editor

Use the Display Editor 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

You can close it multiple ways as well:

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

6.1 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 on the Display view.

6.2 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.

6.2.1 Name

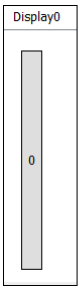
Shows the names of configured displays: Display0, Display1, etc. Each name is unique. It may contain upper or lowercase letters, underscores, and digits. However, the first character cannot be a digit.

6.2.2 Type

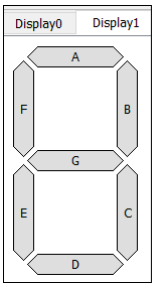
Shows the types of displays to select from the pull-down menu:

- **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.

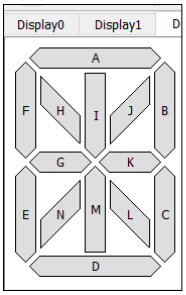
6 Display Editor



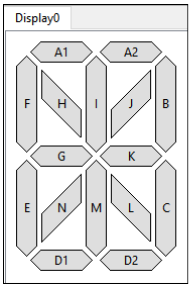
- **7-Segment Display** – Consists of 7 segments per symbol.



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

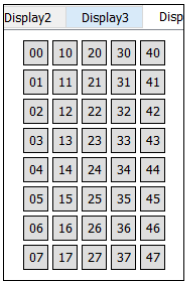


- **16-Segment Display** – Consists of 16 segments per symbol.



***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.

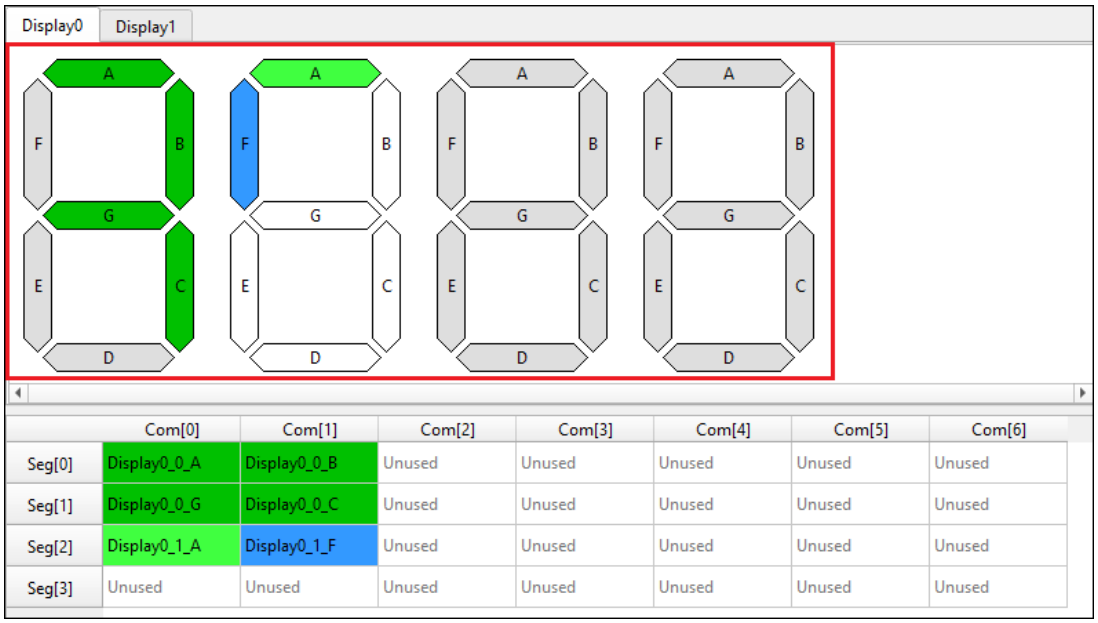


6.2.3 Symbols

The number of display types to include on the corresponding Display view tab.

7 Display view

The Display view contains one or more tabs that show display types configured in the Display Editor.



	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

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:

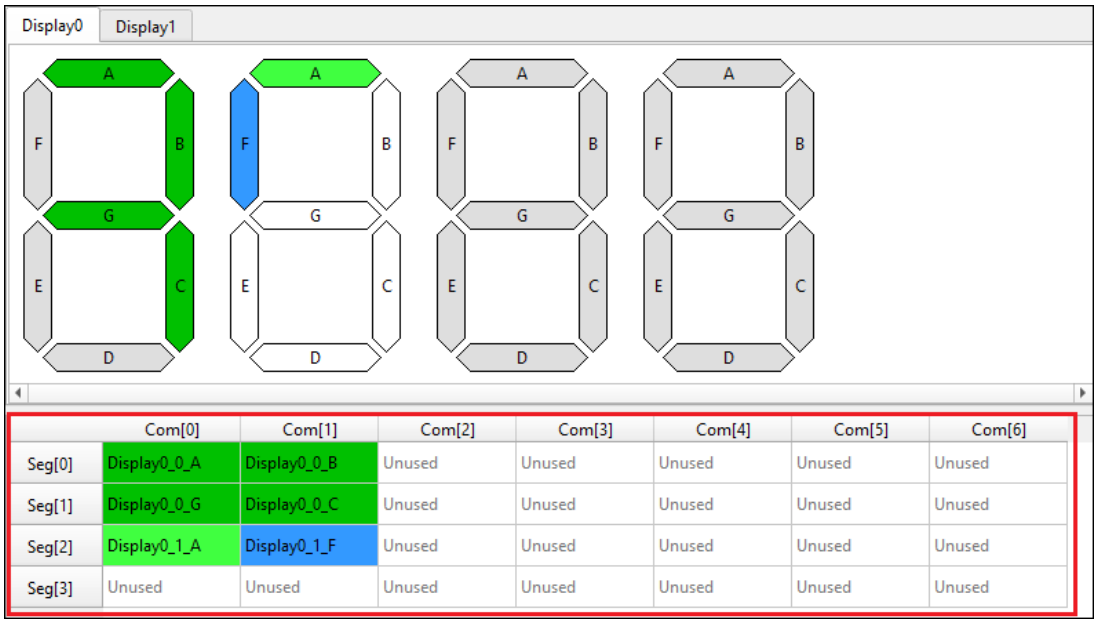
- Light Green – The pixel is connected, and the symbol is currently selected.
- Dark Green – The pixel is connected, and the symbol is currently not selected.
- Blue – The pixel is currently selected (connected or not).
- White – The pixel is not connected, and the symbol is currently selected.
- Grey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

***Note**: The same colors are used for pixels in the mapping table, when applicable.*

***Note**: 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 Abbreviations and definitions under About this document).*

8 Mapping table

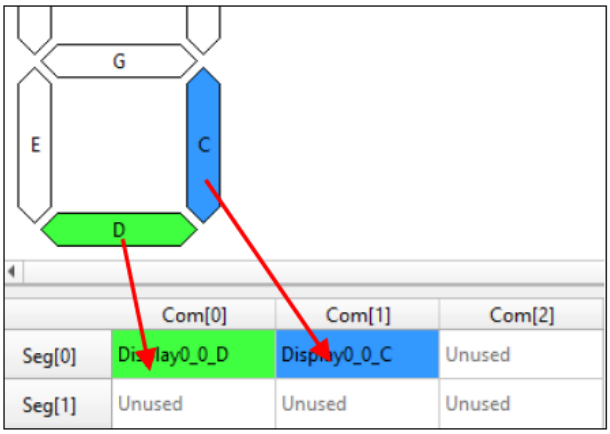
The mapping table (below the Display view) is a grid control where columns represent Commons lines, rows represent Segments lines, and cells represent pixels. Each pixel is determined by a unique pair of common and segment lines.



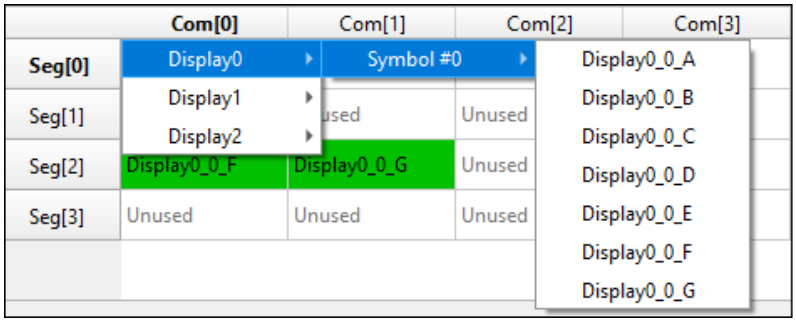
	Com[0]	Com[1]	Com[2]	Com[3]	Com[4]	Com[5]	Com[6]
Seg[0]	Display0_0_A	Display0_0_B	Unused	Unused	Unused	Unused	Unused
Seg[1]	Display0_0_G	Display0_0_C	Unused	Unused	Unused	Unused	Unused
Seg[2]	Display0_1_A	Display0_1_F	Unused	Unused	Unused	Unused	Unused
Seg[3]	Unused	Unused	Unused	Unused	Unused	Unused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

8.1 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.



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



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**.

8.2 Change pixel name

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

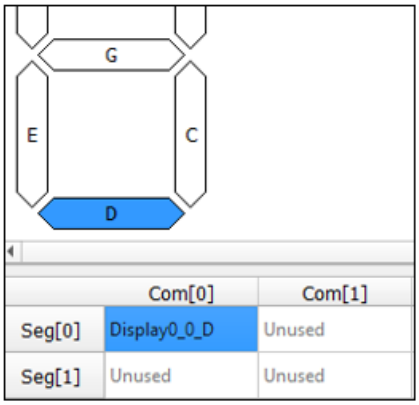
If you name an unconnected pixel from the mapping table, a define is added to the generated header file. For example:

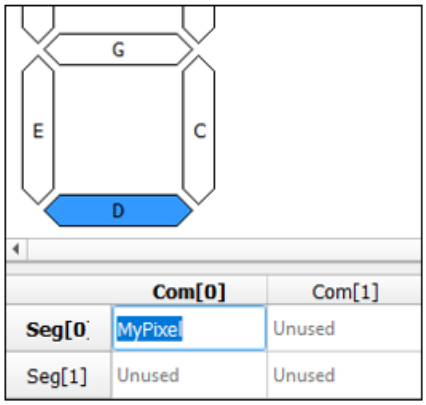
```
#define <Peripheral instance name>_<User defined pixel name> (CY_SEGLCD_PIXEL( <Common
number>UL,  <Segment number>UL))
```

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, delete the name and press [**Enter**].

9 Known issues, limitations, and workarounds

There is a known GUI limitation that you may experience: disconnecting a pixel by simply dragging it outside the mapping table may lead to the case when the cursor movement is not fast enough and the event processing the movement may not be triggered, so the pixel will not disconnect.

To avoid this, do one of the following:

- execute the procedure slowly
- select the pixel and press the [**Delete**] key

10 Version changes

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

Version	Change descriptions
1.0	New tool.
1.10	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{№}" 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.
1.30	Removed the command-line generate options: -g and –generate.
1.40	Updated the GUI by moving to Qt-5.15.2
	Removed: the migration of configuration to the current XML format – configuration saved in the comments in generated HEADER files (the old method).
1.50	Changed the device library file from xml to props.json.
1.51	Fixed bugs.
1.60	Back-end changes.
1.61	Minor back-end changes.
1.70	Minor back-end changes.
1.80	Minor back-end changes.
1.90	Minor back-end changes.

Revision history

Revision	Date	Description
**	2019-10-16	New tool.
*A	2019-11-25	minor changes.
*B	2020-03-27	Updated versioning to 1.10.
*C	2020-09-01	Updated versioning to 1.20.
*D	2020-12-14	Updated versioning to 1.21.
*E	2021-03-11	Updated to version 1.30.
*F	2021-09-22	Updated to version 1.40.
*G	2022-09-28	Updated to version 1.50.
*H	2023-05-18	Updated to version 1.51.
*I	2024-01-25	Updated to version 1.60.
*J	2024-09-27	Updated to version 1.61.
*K	2024-12-06	Updated to version 1.70.
*L	2025-03-21	Updated to version 1.80.
*M	2025-08-29	Updated to version 1.90.

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2025-08-29

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about any aspect of this document?

Email: erratum@infineon.com

Document reference
IFX-cpa1712675479309

Important notice

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

Warnings

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.