# Article information:

STM32 — LVGL documentation  
<https://docs.lvgl.io/master/get-started/platforms/stm32.html>

# Article summary:

1. LVGL can be added to STM32CubeIDE in a similar fashion to any other Eclipse-based IDE.

2. To include LVGL in a project, copy the entire LVGL folder to the project's Drivers/lvgl directory and add the include path in the project settings.

3. The article provides examples of how to use LVGL with STM32CubeIDE, including a bare metal example and a FreeRTOS example.

# Article rating:

Appears moderately imbalanced: The article provides some useful information, but is missing several important points or pieces of evidence that would be required to present the discussed topics in a balanced and reliable way. You are encouraged to seek a more balanced perspective on the presented issues by exploring the provided research topics and looking at different information sources.

# Article analysis:

The article titled "STM32 — LVGL documentation" provides a tutorial on how to integrate the LVGL (Light and Versatile Graphics Library) into STM32CubeIDE, an Eclipse-based IDE for STM32 microcontrollers. The article includes instructions for including LVGL in a project, as well as examples of using LVGL in both bare metal and FreeRTOS environments.

Overall, the article appears to be informative and provides step-by-step instructions for integrating LVGL into STM32CubeIDE. However, there are a few potential biases and missing points of consideration that should be noted:

1. Biases: The article seems to have a bias towards using STM32CubeIDE as the preferred IDE for working with STM32 microcontrollers. While it is mentioned that LVGL can be added to any Eclipse-based IDE, the focus is primarily on STM32CubeIDE.

2. Unsupported claims: The article mentions that LVGL is not thread-safe and suggests using Mutexes when using it with FreeRTOS. However, no evidence or explanation is provided to support this claim. It would be helpful to provide more information on why LVGL is not thread-safe and how Mutexes can address this issue.

3. Missing evidence: The article mentions that the frame buffers need to be created as global variables but does not explain why this is necessary or provide any evidence to support this requirement. It would be beneficial to include an explanation or reference to further reading on this topic.

4. Unexplored counterarguments: The article does not explore any potential drawbacks or limitations of using LVGL with STM32 microcontrollers. It would be useful to include information on any known issues or challenges that users may encounter when working with LVGL in this context.

5. Partiality: The article focuses solely on integrating LVGL into STM32CubeIDE and does not provide information on alternative methods or tools for working with LVGL on STM32 microcontrollers. This could be seen as promoting STM32CubeIDE over other options without considering the potential benefits of alternative approaches.

In conclusion, while the article provides a useful tutorial for integrating LVGL into STM32CubeIDE, it has some biases and missing points of consideration that should be taken into account. It would be beneficial to provide more evidence, explore counterarguments, and present a more balanced view of the topic.

# Topics for further research:

* Thread safety in LVGL library
* Mutexes in FreeRTOS and LVGL integration
* Importance of global variables in LVGL frame buffers
* Limitations of using LVGL with STM32 microcontrollers
* Alternative IDEs for working with LVGL on STM32 microcontrollers
* Comparing STM32CubeIDE with other development tools for LVGL integration

# Report location:

<https://www.fullpicture.app/item/3f519cfaac8818c05feb4ac9afb207dd>