SYMBOLA

Code with GCC & GTK4 on Raspberry Pi

Table of Contents

Disclaimer

This guide is provided for informational purposes only. Every effort has been made to ensure the accuracy and completeness of this guide, but it is provided “as is” without warranty of any kind, express or implied. The author shall not be held liable for any direct, indirect, incidental, or consequential damages or losses arising from the use of this guide.

The procedures and software described in this guide are subject to change and may not be up-to-date. Users are advised to exercise caution and consider their specific circumstances when following the instructions.

This guide may contain links to external websites. The author is not responsible for the content or accuracy of any external site.

Please use this guide responsibly and at your own risk.

Contact the Author

If you've spotted an error or simply wish to make contact, feel free to leave a message at:

symbola.co.uk/contact

Your feedback and inquiries are always welcome!

Preface

Introduction to GTK

GTK, which stands for GIMP Toolkit, is a highly versatile library for creating graphical user interfaces (GUIs). Originally developed for GIMP (GNU Image Manipulation Program), GTK has evolved into a widely used, open-source framework for developing GUI applications across a range of platforms, including Linux, Windows, and macOS. GTK is written in C, but it has been designed from the ground up to support a wide variety of languages through its bindings, making it accessible to developers working in Python, C++, Rust, and many others.

GTK follows the principles of the GNOME project, focusing on providing a clean, accessible, and user-friendly interface for desktop applications. Over the years, GTK has undergone several major revisions, each introducing significant improvements and changes to its architecture, features, and design philosophy.

Preface on GTK 4

GTK 4 is the latest major version of the toolkit, released after several years of development and improvements over its predecessor, GTK 3. GTK 4 introduces numerous enhancements and new features aimed at making GUI development more efficient and flexible.

Introduction to the Raspberry Pi 5 (8GB model)

The Raspberry Pi is a compact, affordable computer that has transformed digital learning and hobbyist projects worldwide. Originally designed to teach computer science basics, it has become a favorite for a wide range of users from educators and students to DIY enthusiasts and professional developers. Its small size, low cost, and considerable computing power make it ideal for various applications, including educational projects, home automation, and even commercial product development.

Featuring an ARM-based CPU, RAM, USB ports, GPIO pins for hardware interfacing, HDMI output, and network connectivity, all on a board roughly the size of a credit card, the Raspberry Pi stands out for its versatility. Its GPIO pins, in particular, enable connections to external devices and sensors, making it perfect for robotics and Internet of Things (IoT) applications.

Raspberry Pi OS, the official operating system, offers a user-friendly desktop experience with preloaded software for programming and multimedia, making it accessible to beginners. With numerous versions improving on processing power and connectivity, the Raspberry Pi continues to support a mission of democratizing access to computer science education, providing a platform for users to explore computing and realize creative projects.

What to Expect from This Guide

This guide is designed to walk you through the essentials of setting up your Raspberry Pi 5, from hardware assembly to software installation, and introduces you to programming with GTK 4 using Visual Studio Code. You'll find straightforward steps to get your Raspberry Pi up and running, followed by detailed instructions on how to set up Visual Studio Code as your development environment for GTK 4 applications. We'll also cover the basics of creating a simple GTK 4 program that opens a window, providing a solid foundation for your future programming projects with GTK 4 on the Raspberry Pi.

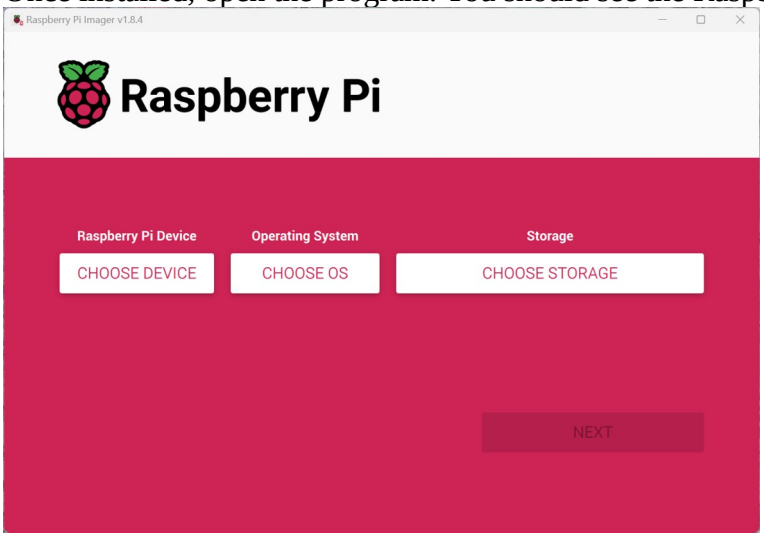
Note: This guide specifically utilizes a Raspberry Pi 5 (8GB) model with Raspberry Pi OS 64-bit, ensuring a smooth and optimized experience for both setup and programming with GTK 4 using Visual Studio Code.

Hardware Requirements for Setting Up Your Raspberry Pi

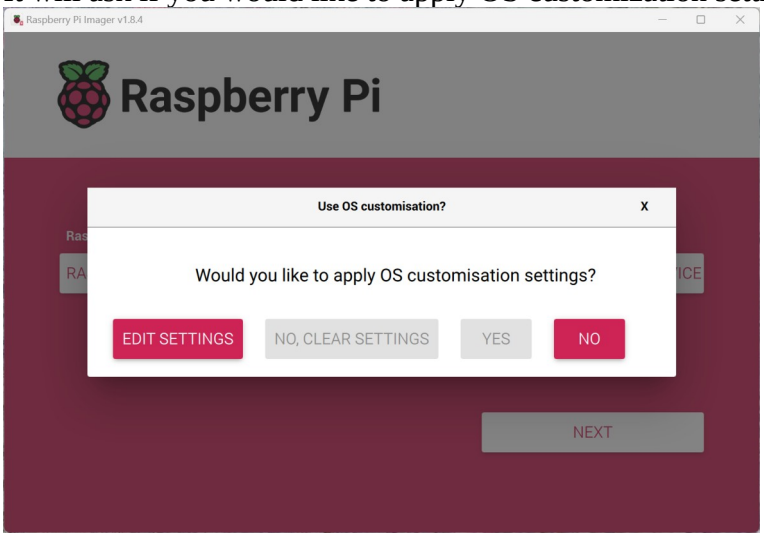
1. **Raspberry Pi (Raspberry Pi 5 8GB used in this guide)**
   This guide uses the Raspberry Pi 5 (8GB) for its optimal performance and features. However, older models such as the Raspberry Pi 4 or 3 are also compatible but may result in slower performance. The Raspberry Pi 5 with 8GB RAM is recommended for the best experience, though models with 4GB or 2GB of RAM can also be used, with an understanding that performance may be impacted.
2. **Power Supply**
   An appropriate and reliable power supply for the Raspberry Pi 5 is crucial for stable operation. The official power supply for this model is highly recommended. Make sure to purchase the correct version for your country!
3. **MicroSD Card**
   At least 16GB recommended, with a preference for 32GB or more for better storage and performance. Use a class 10 microSD card for improved read/write speed.
4. **Keyboard and Mouse**
   Essential for the initial setup and configuration.
5. **Monitor/Display**
   A monitor with an HDMI interface is required for the Raspberry Pi.
6. **Micro-HDMI to HDMI Cable**
   This cable is needed to connect the Raspberry Pi to the display.
7. **Ethernet Cable or Wi-Fi Connectivity**
   Internet access is crucial for downloading necessary software and updates.
8. **Cooling Fan**
   To ensure the best performance, a cooling fan is essential for this model.
9. **Case for Raspberry Pi (Optional)**
   A case is recommended to protect the hardware.
10. **MicroSD Card Reader**
    Either a built-in or an external USB reader will suffice.
11. **Access to Another Computer**
    This is necessary to install the operating system files onto the MicroSD card. This can be a Windows or Mac computer.

Writing the Operating System to the MicroSD card

1. **Install Raspberry Pi Imager on Your Computer**
   Download it from: raspberrypi.com/software.
2. **Insert the MicroSD Card into the Reader**
   Place your MicroSD card into the card reader, then connect it to your computer.
3. **Install and Run the Raspberry Pi Imager**
   Once installed, open the program. You should see the Raspberry Pi Imager interface.

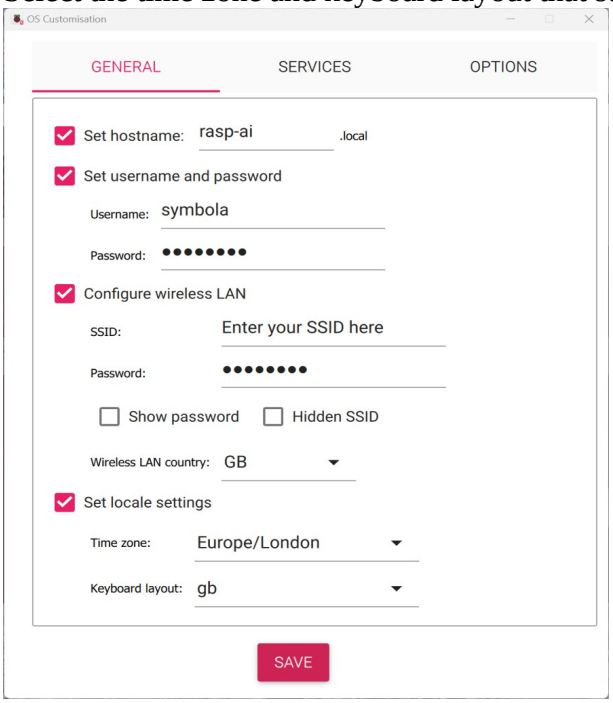


4. **Choose Raspberry Pi Device**
   Select "Raspberry Pi 5."
5. **Choose Operating System**
   Select "Raspberry Pi OS (64-BIT)."
6. **Choose Storage (The MicroSD Card to Write To)**
   Note: All data on the MicroSD card will be deleted!
   Choose your MicroSD card from the list.
7. **Click “Next”**
8. **A Pop-Up Message Box Will Appear**
   It will ask if you would like to apply OS customization settings.

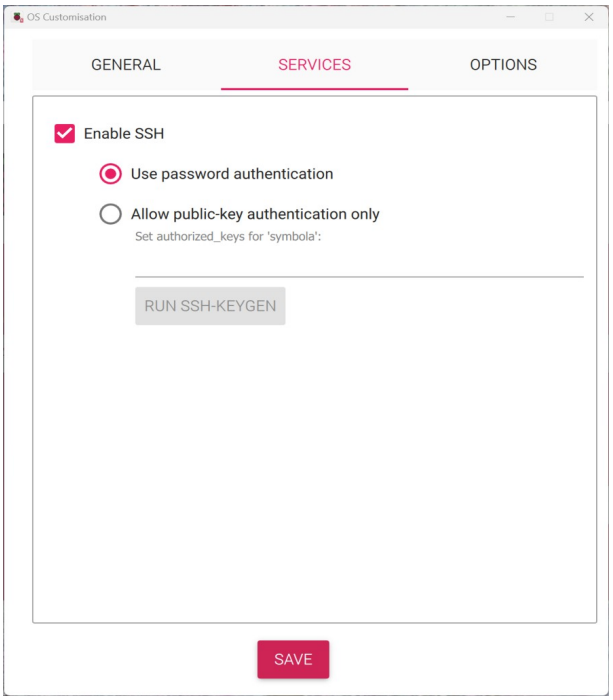


9. **Select “Edit Settings”**

10. **Set Hostname**
    Enter "rasp-ai" as the hostname. (*This was named for another guide; here, you could use "rasp-gtk" or anything you wish*).
11. **Set Username and Password**
    Don’t forget these credentials!
12. **Configure Wireless LAN**
    If you’re unsure of your Wi-Fi network's SSID, check on your phone, laptop, or Wi-Fi router.
13. **Set Locale Settings**
    Select the time zone and keyboard layout that suits your setup.



14. **Enable SSH in the 'Services' Tab**



15. **Leave Options in the 'Options' Tab As Is**
16. **Click on “Save”**

17. **Back to the Pop-Up Message Box**
    When asked again about applying OS customization settings, select "YES".
18. **Read the Notice About Data Erasure**
    Select "YES" (only if you are okay with all data being erased on the MicroSD card).
19. **Wait for the Operating System to be Written to the MicroSD Card**

Setting up the hardware

1. **Insert the Prepared MicroSD Card into Your Raspberry Pi**
2. **Fit the Cooling Fan** (if purchased)
   Before powering up your Raspberry Pi, attach the cooling fan. This step is crucial to maintain optimal operating temperatures, especially important when running demanding applications. Follow the instructions provided with your cooling fan for proper installation.
3. **Place the Raspberry Pi in a Case** (if purchased)
   If you have a case for your Raspberry Pi, now is the time to secure the board inside it. A case helps protect the Raspberry Pi from physical damage and dust. Make sure that the case is compatible with the cooling fan setup and allows proper ventilation.
4. **Connect to a Display**
   Use the Micro-HDMI to HDMI cable to connect your Raspberry Pi to the monitor or display.
5. **Attach Keyboard and Mouse**
   Connect these peripherals to the USB ports on the Raspberry Pi.
6. **Ethernet Connection (Optional)**
   If you are not using Wi-Fi, connect your Raspberry Pi to the network using an Ethernet cable. This will provide a more stable internet connection, which is crucial for downloading updates and software.
7. **Connect to Power Supply**
   Plug in the power supply to your Raspberry Pi. Ensure you are using the recommended power adapter to prevent any power issues.
8. **Ensure Secure Connections**
   Before powering up, double-check that all connections (display, keyboard, mouse, Ethernet, and power supply) are secure. A loose connection could lead to problems during the setup process.
9. **Power On the Raspberry Pi**
   Once everything is connected and secured, power on your Raspberry Pi. Proceed with the initial setup of the Raspberry Pi OS.

First Boot

Upon the first boot, after the desktop screen loads, a pop-up window titled “Wi-Fi Network Authentication Required” may appear. If it does, select “Connect” to establish an internet connection.

If all has gone well, your Raspberry Pi should boot up to the desktop and be connected to the internet. At the top left of the screen, from left to right, you will see several icons:

1. **The Raspberry Pi Logo**
   This is the operating system's main "Start" menu, where you can access programs, settings, etc.
2. **The Globe Icon**
   This represents the Chromium web browser by default.
3. **Two Yellow Folders**
   These are icons for the file explorer, where you can manage and access your files.
4. **The Dark Box with “>_”**
   This is the terminal application, a powerful tool for running various commands and scripts.

On the right side at the top, you'll find icons for Bluetooth, Wi-Fi, volume control, and the time display. These are quick access buttons that allow you to manage your device's connectivity and settings easily.

Open a Terminal Window

Much of the installation process will be conducted using the terminal. Here's how to open it:

1. **Navigate to the Terminal**
   Look at the top left of the screen and find the icon that looks like a dark box with “>_” symbol. This is the terminal application.
2. **Open the Terminal**
   Click on the “>_” icon. A terminal window will open, displaying a command prompt, which is where you can enter your commands.
3. **Using the Terminal**
   Once the terminal window is open, you will see a command prompt waiting for commands to be entered.

Tip: If you are viewing this guide as a PDF document on your Raspberry Pi, you can conveniently copy and paste the commands directly into the terminal. This can save time and reduce the chance of typing errors.

Check for system updates

Update the System. All these commands need to be typed (or copied and pasted) into an open Terminal window. (Commands shown in **bold** and **green**).

1. Update package list
   - **sudo apt update**
2. Upgrade installed packages
   - **sudo apt full-upgrade**

Install Visual Studio Code

1. **Download VS Code** from:
   - https://code.visualstudio.com/Download
   - Download: Arm64 .deb
2. **Alternatively, go directly to**:
   - https://code.visualstudio.com/docs/?dv=linuxarm64_deb
3. **This will download the file**:
   - code_1.87.0-1709077383_arm64.deb
   - **Note:** The file you download might have a different filename due to updates. If so, update the filename in the instructions to the one you have downloaded.
4. **Open the Terminal and change the directory to Downloads**:
   - **cd Downloads/**
5. **Install VS Code**:
   - **sudo apt install ./code_1.87.0-1709077383_arm64.deb**
6. **Launch VS Code**:
   - Type **code** in the terminal to launch VS Code.
   - Or launch it from the ‘Start’ menu.

Install development tools

1. **Install GTK 4 development libraries**:
   - **sudo apt install libgtk-4-dev -y**
2. **Install essential build tools**:
   - **sudo apt install build-essential -y**
3. **Install the GNU Debugger (GDB)**:
   - **sudo apt install gdb -y**

Setting up the new project & First run

Note: Where "**/home/symbola/**" is shown, replace "**symbola**" with the username you chose when setting up Raspberry Pi OS.

1. **Download the source code**:
   - Visit: https://github.com/SYMBOLA2024/SYM7
   - Click on the green "Code" button and select “Download ZIP”.
2. **Extract the ZIP file** (filename: SYM7-main.zip):
   - Right-click on the file in the Downloads folder and select "Extract Here".
3. **Create a folder called "Code"**:
   - /home/symbola/Code
4. **Create a folder called "HelloGtk"**:
   - /home/symbola/Code/HelloGtk
5. **Create a folder called ".vscode"** (note the “.” in front of vscode):
   - /home/symbola/Code/HelloGtk/.vscode
   - The folder “.vscode” will be hidden. To show it, right-click in the File Manager and select "Show Hidden".
6. **From the extracted ZIP file, copy**:
   - “**main.c**” & “**Makefile**”
   - **To**: /home/symbola/Code/HelloGtk
7. **From the extracted ZIP file (vscode folder), copy**:
   - “**c_cpp_properties.json**”, “**launch.json**” & “**tasks.json**”
   - **To**: /home/symbola/Code/HelloGtk/.vscode
8. **Launch Visual Studio Code**:
   - Through the OS start menu or terminal
9. **File ► Open Folder**:
   - /home/symbola/Code/HelloGtk
   - Tick the box to trust the authors.
10. **Open the main.c file in the IDE**:
    - When asked, install the “C/C++ Extension Pack” in Visual Studio Code.
11. **Run the code**:
    - Ensure the **main.c** code file is visible (select the **main.c** tab if required).
    - In the menu, select: Run ► Start Debugging (or press F5).
    - After a short while, the program will launch, displaying an empty GTK4 window.
    - Close the window.
12. **Test the debugger with a breakpoint**:
    - To the left of line 12, click the mouse button. A red dot should appear; this is a breakpoint.
    - In the menu, select: Run ► Start Debugging (or press F5).
    - The debugger will stop at line 12, a yellow arrow will surround the red breakpoint dot.
    - Click continue on the debugger control (small blue arrowhead pointing right) (or press F5).
    - The program will continue to run and show the window.
13. **The use of the debugger and breakpoints can be very useful in finding bugs and in larger projects to help understand the flow of the program**.

Conclusion

In conclusion, this document has provided a very brief overview and a getting-started guide aimed at introducing you to setting up and programming with GTK 4 on a Raspberry Pi 5 using Visual Studio Code. While we've covered the essentials to get you up and running, it's important to delve deeper into understanding the setup files, such as the “Makefile” and the files within the “.vscode” folder. These files play a crucial role in the development process, affecting how your projects are built and debugged.

For more comprehensive information on using GTK 4 and Visual Studio Code, you are encouraged to visit the project's web homepages. These resources offer extensive documentation, tutorials, and community support that can help you expand your knowledge and skills.

Additionally, engaging with the wider developer community through forums, social media, and online platforms can provide valuable insights, tips, and best practices. Experimenting with different aspects of GTK 4 and Visual Studio Code, contributing to open-source projects, and building your own applications are excellent ways to enhance your learning experience.

Remember, the journey into software development is ongoing, and there's always something new to learn. This guide serves as your springboard into the fascinating world of GUI application development with GTK 4 on the Raspberry Pi. Embrace the challenges and opportunities that come your way, and enjoy the process of creating something remarkable. Happy coding!